Autonomous Sailboat Navigation

Novel Algorithms and Experimental Demonstration

Roland Stelzer

Centre for Computational Intelligence

De Montfort University, Leicester

A thesis submitted as partial fulfilment of
the requirements for the degree of

Doctor of Philosophy

2012

I would like to dedicate this thesis to Dr. Herbert Hörtlehner, whose enthusiasm and unconventional way of education has awaken my interest in robotics and artificial intelligence. Herbert had been on my supervisory team since he unexpectedly died in September 2005. May he rest in peace.

Acknowledgements

Firstly, I would like to thank my supervisors Prof. Robert John, Dr. Mario Gongora and Dr. Tobias Pröll, who always had time to answer my questions and have repeatedly supported me with their helpful advice. I would also like to express my special thanks to my unofficial supervisors Dr. Jenny Carter and Dr. Simon Coupland for their many useful tips on how to structure my work, and to Peter Farnell for his linguistic corrections.

Above all I would like to thank the Austrian Society for Innovative Computer Sciences (INNOC). With the support of INNOC, I was able to carry out my research under optimal conditions. By name I should like to thank Karim Jafarmadar from INNOC, who helped me in my research into autonomous sailing boats from the outset. I should also like to express my thanks to all of the many other members of INNOC who actively supported me and my research project.

Furthermore, I would like to thank the Austrian Federal Ministry of Science and Research. The project is partially realised within the funding programme *Sparkling Science*.

Finally, I would like to thank my family for the patience that they have shown with me and my work. Thank you, Silvia, Elena and Jonathan!

Abstract

The purpose of this study was to investigate novel methods on an unmanned sailing boat, which enables it to sail fully autonomously, navigate safely, and perform long-term missions.

The author used robotic sailing boat prototypes for field experiments as his main research method. Two robotic sailing boats have been developed especially for this purpose. A compact software model of a sailing boat's behaviour allowed for further evaluation of routing and obstacle avoidance methods in a computer simulation. The results of real-world experiments and computer simulations are validated against each other.

It has been demonstrated that autonomous boat sailing is possible by the effective combination of appropriate new and novel techniques that will allow autonomous sailing boats to create appropriate routes, to react properly on obstacles and to carry out sailing manoeuvres by controlling rudder and sails. Novel methods for weather routing, collision avoidance, and autonomous manoeuvre execution have been proposed and successfully demonstrated. The combination of these techniques in a layered hybrid subsumption architecture make robotic sailing boats a promising tool for many applications, especially in ocean observation.

Contents

List of Figures

List of Tables

Nomenclature

Nautical Terms

America's Cup The America's Cup is a trophy awarded to the winner of the America's Cup match race between two yachts. The America's Cup is the oldest active trophy in international sport.

Apparent Wind Apparent wind is referred to as the velocity of air as measured from a moving object, such as a ship.

Autohelm Autohelm is a Raymarine trademark, but often used generically.

Automatic Identification System Automatic identification system (AIS) is a navigation system for locating, identifying and tracking marine vessels. Maritime laws require AIS on voyaging ships with gross tonnage of 300 or more.

Beating Beating to windward is referred to as the process of zigzagging when sailing upwind.

Great Circle Great circle route is the shortest route between two points on the surface of a sphere, e.g. the earth.

Heeling Heeling is the sidewards tilt of a sailing boat usually caused by lateral wind force.

Jib A jib (also spelled jibb) is a sail set ahead of the mast of a sailing vessel.

Jibe	A jibe (also referred to as jib or gybe) is when a sailing boat turns its stern through the wind, such that the direction of the wind changes from one side of the boat to the other.
Leeway	Leeway is the amount or angle of the drift of a ship to leeward from its heading.
Luffing	It's called luffing when the sail flaps in the wind.
Mainsail	A mainsail (also just main) is a sail located behind the main mast of a sailing vessel.
Points of Sail	Point of sail describes the direction of a boat with regard to the direction of the wind (see Figure 2.5).
Rigging	Rigging is the mechanical sailing apparatus attached to the hull in order to move the boat as a whole. This includes cordage, sails, and spars (masts and other solid objects sails are attached to)
Schooner	A schooner is a type of sailing vessel characterized by the use of two or more masts with the forward mast being no taller than the rear masts.
Single-handed Sailing	Single-handed sailing is sailing with only one crew member. The term is usually used with reference to ocean and long-distance sailing.
Tack	A tack or coming about is the manoeuvre by which a sailing boat or yacht turns its bow through the wind so that the wind changes from one side to the other.
True Wind	The velocity of air as measured from a platform fixed to the ground is known as true wind.
Velocity Made Good	The speed of a sailing boat relative to the waypoint T it wants to reach is also referred to as velocity made good (VMG)

NOMENCLATURE

Acronyms

3G	3rd Generation mobile telecommunications
AAA	All Around Array
AI	Artificial Intelligence
AIS	Automatic Identification System
ANN	Artificial Neural Networks
ASC	Autonomous Surface Craft
ASV	Autonomous Surface Vehicle
AUV	Autonomous Underwater Vehicle
CAN	Controller Area Network
CHR	Constraint Handling Rules
FIS	Fuzzy Inference System
FL	Fuzzy Logic
GPRS	General Packet Radio Service
GPS	Global Positioning System
IMT	International Mobile Telecommunications
IOM	International One Meter class
IRSC	International Robotic Sailing Conference
LOCS	List Of Current Scans
PID	Proportional Integral Derivative
SBD	Short Burst Data (Iridium Service)
UMTS	Universal Mobile Telecommunications System

VMG	Velocity Made Good
VPN	Virtual Private Network
WLAN	Wireless Local Area Network
WRSC	World Robotic Sailing Championship

Publications

During the course of this research project, a number of publications have been made which are based on the work presented in this thesis. They are listed here for reference in descending chronological order.

- Stelzer, R.; Jafarmadar, K. (2011): History and Recent Developments in Autonomous Sailing, in Proceedings of International Robotic Sailing Conference, pp. 3–23, Lübeck, Germany.
- Dabrowski, A.; Busch, S.; Stelzer, R. (2011): A Digital Interface for Imagery and Control of Navico/Lowrance Broadband Radar, in Proceedings of International Robotic Sailing Conference, pp. 169–182, Lübeck, Germany.
- Langbein, J.; Stelzer, R.; Frühwirth, T. (2011): A Rule-Based Approach to Long-Term Routing for Autonomous Sailboats, in Proceedings of International Robotic Sailing Conference, pp. 195–204, Lübeck, Germany.
- Stelzer, R.; Jafarmadar, K.; Hassler, H.; Charwot, R. (2010): A Reactive Approach to Obstacle Avoidance in Autonomous Sailing, in Proceedings of International Robotic Sailing Conference, pp. 34–40, Kingston, Ontario, Canada.
- Klinck, H.; Stelzer, R.; Jafarmadar, K.; Mellinger, D.K. (2009): AAS Endurance: An autonomous acoustic sailboat for marine mammal research, in Proceedings of International Robotic Sailing Conference, pp. 43–48, Matosinhos, Portugal.

PUBLICATIONS

- Stelzer, R.; Jafarmadar, K. (2009): Communication Architecture for Autonomous Sailboats, in Proceedings of International Robotic Sailing Conference, pp. 31–36, Matosinhos, Portugal.
- Stelzer, R.; Pröll, T. (2008): Autonomous Sailboat Navigation for Short Course Racing, in Elsevier Journal of Robotics and Autonomous Systems, Vol. 56 (7), pp. 604–614.
- Stelzer, R.; Pröll, T.; John, R.I. (2007): Fuzzy Logic Control System for Autonomous Sailboats, in Proceedings of IEEE International Conference on Fuzzy Systems, pp. 97–102, London, United Kingdom.
- Stelzer, R.; Jafarmadar, K. (2007): A Layered System Architecture to Control an Autonomous Sailboat, in Proceedings of TAROS 2007, pp. 153–159, Aberystwyth, Wales, UK.
- Stelzer, R.; Jafarmadar, K. (2007): Simple Communication Protocol for Rapid Robot Prototyping, in Proceedings of Humanoid and Service Robotics Conference, pp. 149–156 , Kosice, Slovakia.

Chapter 1

Introduction

This thesis reports the results of research into robotic sailing. In particular it is concerned with evaluation of existing methods for all areas of robotic sailing as well as conception, development, demonstration, and critical evaluation of novel methods where necessary and appropriate. This research proposes novel algorithms to short course routeing, reactive collision avoidance and execution of sailing manoeuvres on an unmanned sailing vessel without any human intervention.

This chapter provides an introduction into the topic of robotic sailing. After a definition of robotic sailing it presents the motivation for this research by presentation of the strengths of unmanned autonomous sailing boats and leads over to potential applications. The aims and objectives of this research are discussed and stated in the research hypothesis and the research questions. The organisation of the remainder of the thesis is also given after a summary of the contributions to knowledge to the field of robotic sailing.

1.1 Robotic sailing

By robotic sailing we mean that the whole process of sailing boat navigation is performed by an autonomously acting system of technical devices. Bowditch [2010] defines navigation as *"the process of monitoring and controlling the movement of a craft or vehicle from one place to another"*.

Robotic sailing boats therefore have to perform the complex planning and manoeuvres of sailing fully automatically and without human assistance. Starting off by calculating an optimum route based on weather data and going on to independent tacking[1] and jibing[2] and avoiding collisions, stand-alone sailing boats are able to sail safely and reliably through to any and every destination. The human being merely has to enter the destination co-ordinates.

The key characteristics of a robotic sailing boat can be summarized as follows:

- Wind is the only source of propulsion.
- It is not remote controlled; the entire control system is on board.
- It is completely energy self-sufficient; this is not a must in the sense of definition of a robotic sailing boat, but it opens a wide range of applications.

1.2 Motivation

Many technical aids, such as self-steering gears (see Section 2.1.1 for details), chartplotters[3], electric winches, or weather routing software are available for common sailing boats. However, relatively little time and effort has been spent on autonomous sailing. Research on autonomous surface vehicles (ASV) has been mainly focused on short-range crafts powered by electric or combustion engines. Such crafts are limited in range and endurance depending on the amount of fuel or battery capacity on board to run a motor for propulsion. In contrast a sailing vessel needs only a minimal amount of power to run sensors, computers and to adjust sail and rudder position.

Recent events, like the devastating tsunami in Asia in 2004, the Deepwater Horizon oil spill in Gulf of Mexico in 2010, accidents involving refugee boats off the coast of Lampedusa, Italy, and pirate activities in the Gulf of Aden have

[1] A tack or coming about is the manoeuvre by which a sailing boat or yacht turns its bow through the wind so that the wind changes from one side to the other.

[2] A jibe (also referred to as jib or gybe) is when a sailing boat turns its stern through the wind, such that the direction of the wind changes from one side of the boat to the other.

[3] A Chartplotter is a device that displays an electronic navigational chart along with the position, heading and speed of the boat, and may display additional information from radar or other sensors.

clearly emphasized impressively the importance of a fully integrated ocean observation system (Rynne & von Ellenrieder [2009]). Autonomous underwater vehicles (AUV) and motorised ASVs have been widely-used for ocean observations for many years (Bertram [2008]; Rynne & von Ellenrieder [2009]).

A principal problem in ocean monitoring is the limitation in spatial and temporal coverage of the observations (see Fig. 1.1). Measurement can either be done with a moving platform (e.g. research vessel) or stationary recording devices (e.g. anchored recorders). Moving platforms offer the possibility of sampling a large area in a short period of time. However, because of the high costs of ship time temporal coverage is very limited. In contrast, stationary recording devices allow continuous sampling of an area. Their disadvantage lies in the limited spatial coverage of the devices (Mellinger *et al.* [2007]).

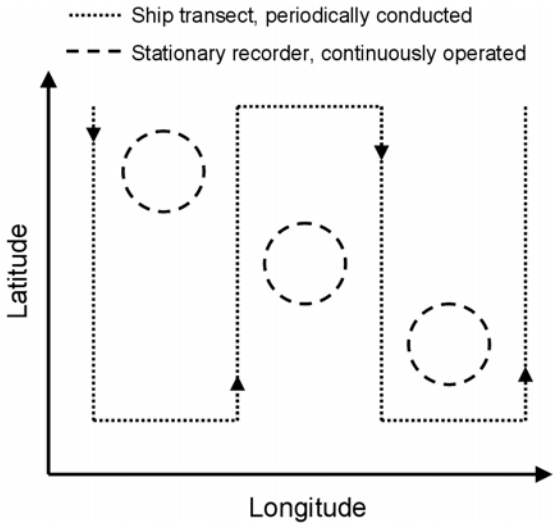


Figure 1.1: Comparison of the spatial and temporal coverage of a research vessel (dotted line) and stationary recorders (dashed circles)

Autonomous and remotely navigable ocean observation platforms offer the possibility of sampling an area of interest with high temporal and spatial resolution at low cost. To date, two autonomous and remotely navigable platforms are available for research on the ocean: wave-powered vessels (e.g. the Wave Glider^{TM}, Liquid Robotics [2009]) and ocean gliders (e.g. the Seaglider^{TM}, Eriksen *et al.* [2001]).

The Wave Glider provides a submerged (swimmer) and a surface (float) unit. Both units are connected via a tether and allow the swimmer to move up and down as a result of wave motion. The swimmer includes several fins which interact with the water as the swimmer moves up and down, and generate forces which propel the vehicle forward. The Wave Glider, developed by Liquid Robotics, Inc., has proven long-term capabilities in a five-month test trial, and the device seems well-suited for long-term observations.

Gliders are commercially available from several manufacturers (e.g. Seaglider [2009]), and all types are based on the same principle. Changes in buoyancy cause the glider to move down and up in the water, and as with an aeroplane glider, wings transform this vertical motion into forward motion. A stable, low-drag, hydrodynamic shape allows the glider to fly efficiently through the oceans. These devices are optimized for extremely low energy requirements and designed to operate at depths up to 1000 m. Gliders are capable of long-term operation and have been used extensively for oceanographic research for a number of years.

An advantage of submerged operated vehicles is the limited surface time, which minimises the risk of a collision with other obstacles, reduces damage from high-energy surface phenomena (wind and waves), and reduces the possibility of potentially harmful human action. Furthermore gliders can be deployed in polar regions, where ice coverage prohibits the usage of surface vehicles, and in areas with high wind and waves where the traditional visual means of marine mammal observation are ineffective. On the other hand, submerged operated platforms such as gliders also suffer from some drawbacks:

- **Speed:** The typical horizontal cruise speed of most gliders is approximately 0.25 m/s (0.5 kt). This low speed does not allow surveying a large area in a reasonably short time period. To be able to conduct a survey in a shorter amount of time, a larger number of gliders (number depending on the size of the area of interest) must be deployed. A larger number of devices significantly increases the complexity and cost of a survey.

- **Payload:** Most gliders are relatively small instruments and provide relatively limited payload capacity. Larger payloads allow for more batteries and sensors, so the small capacity of gliders limits both their deployment

duration and their capability for measuring a wider suite of oceanographic parameters. An additional constraint in gliders is that the payload must be horizontally balanced.

- **Continuous real-time access:** As gliders stay submerged most of time, these platforms do not provide continuous real-time access. For real-time monitoring the minimum response time of a glider is the time it takes to rise to the surface - potentially several hours - plus a small amount of data transmission time.

- **Sensors:** The operating power for gliders comes from batteries. Because of constraints in payload mass, the amount of energy available for operating power-intensive electronics such as optical sensors is small.

- **Computational power:** Because of the energetic limitations, sophisticated and thus energy-intensive computations cannot be run continuously onboard a glider.

- **Reliability:** A malfunction at depth can cause the loss of a glider.

- **Duration:** Because of the limited energy capacity, glider deployments for long-term studies are limited to a duration of several weeks.

With an unmanned, autonomous, and energy self-sufficient robotic sailing boat, it is possible to overcome many of the disadvantages and limitations of today's technologies for ocean monitoring.

1.3 Potential applications

Beside ocean monitoring a few more applications are possible. However, not all of the following applications are likely to be realised within the next few years.

- **CO_2-neutral transportation of goods and unmanned ferrying:** The price of fuel is expected to increase dramatically in the next few decades and additionally, penalties for CO_2 emissions might add to transport costs. Therefore better alternatives for the transportation of goods or people need

to be sought. Traditional sailing boats are environment-friendly but they require a rather large input in terms of human intervention and therefore incur high personnel costs.

- **Reconnaissance and surveillance:** An autonomous sailing boat can be sent out to remote areas or dangerous regions. Due to its silent, unmanned and energy self-sufficient attributes it is a safe alternative for surveillance of critical areas (piracy, smugglers, fisheries, etc.).
- **Supply vessel:** Secluded regions with a low number of inhabitants or research base camps on islands can be cost-effectively supplied by autonomous sailing boats with equipment, medicine, food or correspondence.
- **Minefield mapping:** Unmanned vehicle systems are useful in their ability to remove humans from dangerous environments. Unmanned robotic sailing boats can explore hazardous regions on the water without exposing people to risks.

1.4 Research hypothesis and research questions

The general aim of the presented work is research in novel methods on an unmanned sailing boat, which enables it to sail fully autonomously, navigate safely (avoid collisions), and perform long-term missions (self-sufficient in terms of energy).

The methods will be proposed in such a way, that minimal adaptations have to be made to a conventional sailing boat. It is not the aim of the work to reinvent sailing or to develop a new sailing boat type, but enable a computer to sail a common sailing boat.

The research hypothesis addressed in this thesis can be stated as:

> *Autonomous boat sailing is possible by the effective combination of appropriate new and novel techniques that will allow autonomous sailing boats to create appropriate routes, to react properly on obstacles and to carry out sailing manoeuvres by controlling rudder and sails.*

This research hypothesis can be broken down into specific research questions with regard to the individual areas of robotic sailing.

- *How can an optimum route to any given way point be determined considering locally measured wind data and wind forecasts? What are the differences between short course and long term routing?*
  The presented work focuses on ship routing optimised in terms of minimum passage time. This is trivial for motorised vehicles in isotropic, stationary environments, where a straight line to the target represents the optimum route. This is significantly different for sailing boats, where the direct line may not be navigable when the target is located upwind. For this study, the routing process for large distances is divided into two stages. A strategic long term routing method plans a rough route based on weather forecasts and previously known fixed obstacles. The result of the long term routing is a series of waypoints which split the entire route into short legs. The short course routing then has to find an optimum course and the optimum moment for course corrections based on locally measured sensor data.

- *How can a robotic sailing boat navigate safely and efficiently around obstacles?*
  Reliable obstacle detection and avoidance is an important problem to be solved for long-term unmanned and autonomous missions on sea. Fixed obstacles such as landmasses can be predefined on the nautical chart which is the basis for the routing system. A combination of multiple techniques, such as thermal imaging, radar, camera, and automatic identification system (AIS) can be used to detect moving obstacles. Research in this field has been carried out for autonomous underwater vehicles (Showalter [2004]) and motorised autonomous surface vehicles (Benjamin *et al.* [2006]; Larson *et al.* [2007]; Smierzchalski [2005]; Statheros *et al.* [2008]). The obstacle avoidance task is different for sailing vessels, as they can not navigate in any direction directly, depending on wind conditions. Therefore a novel approach to autonomous obstacle avoidance is an essential part of this research project.

- *How can human sailors' navigation knowledge and experience be implemented within a computer program? Is fuzzy logic (FL) an appropriate method of keeping an autonomous sailing boat on course and of carrying out sailing manoeuvres? Do the methods work properly on differently sized boats?*
  Both actuators - rudder and sail - should be controlled quickly but smoothly, without jerky leaps and without over-steering. It is investigated, whether this goal can be reached by two Mamdani type (Assilian & Mamdani [1974]) fuzzy inference systems (FIS). Real-world experiments on differently sized sailing robots can demonstrate functionality and scalability of a novel fuzzy logic based control strategy. Moreover, a detailed statistical analysis of log data can test the following hypotheses:

  1. The proposed FL system for rudder control enables an autonomous sailing boat to sail a given heading precisely.
  2. The course deviation is not influenced significantly by the point of sail.
  3. The course deviation is not influenced significantly by the wind speed.

- *How can a reliable data link between boat and shore be guaranteed for the development and future applications of autonomous sailing boats?*
  Although an autonomous sailing boat can operate without human intervention a data link between boat and shore is necessary. During development a reliable connection with high bandwidth and low latency for monitoring, debugging, and remote control in case of emergency is essential. When used for long-term observation the focus is on global network coverage and reliable transmission of a few important values; higher transmission latency can be accepted. The proposed communication system combines multiple data transmission technologies considering network coverage, costs, bandwidth, and latency.

- *What does a flexible, modular and reliable software architecture for autonomous sailing boat control look like? How can existing sailing boat automation devices and methods be combined to allow boats to sail completely autonomously?*

Robot control architectures are usually divided into separate layers, each responsible for a part of the problem. Basically two different architectures exist: deliberative and reactive systems. Whereas deliberative approaches have shown good performance in complex static environments, reactive systems can react quickly in dynamically changing surroundings. The problem to be solved is to find an appropriate control architecture which considers the special requirements for sailing robots and exploits the advantages of both deliberative and reactive approaches.

1.5 Contribution to knowledge

The aim of this work is to identify and to combine existing approaches, as well as to improve them and to introduce novel methods where necessary. Therefore the presented work provides the following contributions to research in robotic sailing:[1]

- Conception, development, simulation, and experimental demonstration of a novel **routing strategy**
- Conception, development, and simulation of a novel reactive approach to **collision avoidance**
- Conception, development, and experimental demonstration of a fuzzy logic **actuator control and manoeuvre execution** strategy
- Conception, design, construction, and experimental demonstration of two **autonomous sailing robots** for demonstration and experimental validation
- Demonstration of **autonomous sailing** on a relevant scale and under relevant conditions

[1] The author has carried out his research in a research group within the Austrian Society for Innovative Computer Sciences (INNOC). However, the contributions to knowledge presented here are his own original work. The following colleagues supported with constructive feedback on the concepts, boat engineering, organising of participation in robotic sailing competitions, and software programming: Sebastian Busch, Raphael Charwot, Adrian Dabrowski, Hannes Hassler, Karim Jafarmadar, Tobias Pröll.

1.6 Contribution to the robotic sailing community

The author contributed actively to the development of a scientific community to promote robotic sailing progress as founder, organiser, and participant of the World Robotic Sailing Championship (WRSC) and the International Robotic Sailing Conference (IRSC).

He participated successfully in many of the international robotic sailing competitions. The author and his team won all of the competitions in which they competed:

- 1st place in Mictrotransat 2006 on a lake in Saint Nicolas de la Grave, Toulouse, France
- 1st place in Microtransat 2007 in the Irish Sea, Aberystwyth, Wales, UK
- 1st place in WRSC 2008 on Lake Neusiedl, Breitenbrunn, Austria
- 1st place in WRSC 2009 on the Atlantic Ocean, Matosinhos, Portugal
- 1st place in WRSC/SailBot 2010 on Lake Ontario, Kingston, Canada
- 1st place in WRSC 2011 on Lake Wakenitz, Lübeck, Germany

The International Robotic Sailing Conference (IRSC) was founded by the author and annually held since 2008: Breitenbrunn, Austria (2008), Matosinhos, Portugal (2009), Kingston, Canada (2010), and Lübeck, Germany (2011). 31 peer-reviewed articles have been published and presented at IRSC so far, six of them from the author's research group. Over the years, the IRSC became an important place for knowledge dissemination in the field of robotic sailing.

1.7 Organisation of the thesis

The succeeding Chapter 2 contains a detailed literature review in the field of robotic sailing. History and scientific community are covered in detail.

Chapters 3–4 present the infrastructure, which has been developed as a basis for further research into robotic sailing techniques:

- **Research design** (Chapter 3): a detailed description of two robotic sailing boats is provided. These prototypes have been developed for field experiments in order to evaluate the novel algorithms which are presented later in this thesis.
- **System architecture** (Chapter 4): a flexible and reliable control and communication infrastructure for robotic sailing boats is presented. Together it represents a framework for all areas of autonomous sailing boat navigation.

Chapters 5–7 present the novelties and contributions to knowledge to the field of robotic sailing in detail. Each of these chapters includes a theoretical description of the proposed approach, an experimental validation, and is concluded by a discussion of the results:

- **Short-course routing** (Chapter 5): a novel method for real-time route optimisation is presented. It relies on locally measured weather data only and reacts immediately on changing wind conditions.
- **Obstacle avoidance** (Chapter 6): a reactive approach is presented, which is an extension to the short course routing method mentioned above. The algorithm enables an autonomous sailing boat to circumnavigate differently sized obstacles under various wind conditions successfully.
- **Manoeuvre execution** (Chapter 7): it is described how basic sailing skills can be transformed into a Mamdani type fuzzy inference system (FIS). The proposed system controls both rudder and sails not just on a straight course, but also during tack and jibe.

Chapter 8 discusses the results and how they have met the original aims. Each of the research questions is evaluated separately. Results are outlined as are their limitations. Finally, this chapter gives an outlook to further research in the field of autonomous sailing.

Chapter 2

Literature review

This chapter gives an overview about history and recent developments in robotic sailing. This includes devices and methods for controlling the rudder and the sails as well as strategies for ship routing. Furthermore advantages and disadvantages of rigid wing sails in comparison to traditional fabric sails are illuminated. Early examples of robotic sailing boats and recent developments, stimulated by robotic sailing competitions such as Microtransat Challenge, SailBot and World Robotic Sailing Championships are presented.

2.1 History of robotic sailing

Extensive research has been undertaken on semi-autonomous systems, where just a subset of the functionality of a robotic sailing boat is covered. The history of self-steering gears and automatic sail control will be discussed independently in the following sections. Afterwards a separate section shows the history of, and recent research projects on completely autonomous sailing.

2.1.1 Self-steering gear

Historically, the first task to be automated was the governing of the rudder. A self-steering gear is an equipment used on ships and boats to maintain a chosen course without constant human action. Self-steering gear is also referred to as

autopilot or *autohelm*[1]. Basically the different forms of self-steering gears can be divided into two categories: mechanical and electronic.

Mechanical self-steering

Fishermen who bind the rudder or tiller of their boat in a fixed position to produce an optimal course can be seen as a first approach to a mechanical self-steering system (Roberts [2008]).

A more sophisticated mechanical approach is the *wind vane* developed first by Herbert "Blondie" Hasler (1914-1987), who is known as one of the fathers of single-handed sailing[2]. Wind vanes are now sold by a number of manufacturers, but most share the same principle: The device consists of a wind vane secured at the stern of the yacht, which is connected to the rudder, specifically a trim tab on the rudder via a system of ropes, pulleys and servos (see Figure 2.1). When the angle of the apparent wind[3] changes, this change is registered by the air vane, which activates the steering device to return the boat to the selected point of sail[4]. Wind vane self-steering does not steer a constant compass course but a constant point of sail.

Electronic self-steering

Electronic self-steering controls the rudder movement by electronics based on various sensor input values. At least a compass is necessary; additional sensors can deliver wind direction or GPS position in order to calculate a heading towards a given target waypoint.

Basically, starting with the development of steering engines on ships attempts were made to control the steering engine based on magnetic compass data. Bennett [1986] reports, that the British Admirality started in the 1860s to equip some

[1] Autohelm is a Raymarine trademark, but often used generically.

[2] Single-handed sailing is sailing with only one crew member. The term is usually used with reference to ocean and long-distance sailing.

[3] Apparent wind is referred to as the velocity of air as measured from a moving object, such as a ship. By contrast, the velocity of air as measured from a platform fixed to the ground is known as true wind.

[4] Point of sail describes the direction of a boat with regard to the direction of the wind (see Figure 2.5).

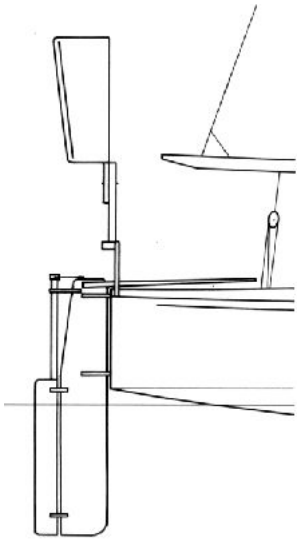


Figure 2.1: Example for a wind-vane with trim tab on main rudder (Scanmar [2011])

of their ships with steering engines in order to carry out manoeuvres much faster. In the 1870s Werner Siemens[1] came up with the idea to control a German Navy torpedo boat automatically. The rudder of the boat was turned by an electric motor which was operated by electromagnet relays. The control could be done either manually via a cable or from the magnetic needle of a compass placed on the boat.

However, it was not until the early 20$^{\text{th}}$ century when steering engines became more common on ships that a number of automatic steering systems were subsequently invented. Sir James Henderson was granted a patent on *automatic steering device* in 1913 which used both heading error and heading error rate in a feedback loop (Roberts [2008]).

Further substantial progress toward automatic steering was based on the invention of electronic gyrocompasses. This helped to overcome the problem of local anomalies in the terrestrial magnetic field. The earliest known gyroscope-like instrument was made and first mentioned by Bohnenberger [1817]. In the 1860s, the advent of electric motors made it possible for a gyroscope to spin indefinitely.

[1]Ernst Werner Siemens (1816-1892) was a German inventor and industrialist. His name has been adopted as the SI unit of electrical conductance, the Siemens. He was also the founder of the electrical and telecommunications company Siemens.

This led to the first prototype gyrocompass. The first functional marine gyrocompass was patented in 1904 by German inventor Hermann Anschütz-Kaempfe (Anschütz-Kaempfe & von Shirach [1904]).

According to Bennett [1986] and Roberts [2008] the major contributions to the development of a practical automatic steering system were made by Sperry Gyroscope Company. Elmer Sperry developed his first automatic ship steering mechanism in 1911 (Allensworth [1999]; Sperry [1922]). Sperry's gyropilot was known as *Metal Mick* as it was capturing much of the behaviour of an experienced helmsman. It compensated for varying sea states using feedback control and automatic gain adjustments. This lead to a first simple adaptive autopilot.

The work of Minorsky [1922] is also regarded as having made key contributions to automatic ship steering. Nicholas Minorsky presented a detailed analysis of a position feedback control. He formulated the specification of a three-term controller, better known as proportional-integral-derivative (PID) controller. By now all big manufacturers of marine electronics offer electronic self-steering systems which keep a boat on a predefined compass course or a heading relative to the wind direction.

Intelligent rudder control

Conventional electronic self-steering systems found on the majority of vessels at sea still employ PID control algorithms to control the heading (Burns [1995]). Van Amerongen [1984] identified two major disadvantages of this type of controller:

1. It is difficult to adjust manually, because the operator usually lacks the necessary insight into control theory.

2. The optimal adjustment varies and is not known by the user. Changing circumstances require manual readjustment of a series of settings.

Due to the highly dynamic and ever-changing environment, artificial intelligence (AI) techniques, like fuzzy logic (FL), artificial neural networks (ANN), and combinations thereof have received considerable attention with regard to rudder control on ships. Various publications have shown the suitability of FL for rudder

control (Abril *et al.* [1997]; Yeh & Bin [1992]; Zirilli *et al.* [2002]). Polkinghorne *et al.* [1995] furthermore made a comparison of their FL implementation to its conventional PID controlled equivalent. The experiments have shown a much smoother rudder action for the FL controller.

The author again demonstrates a reasonable performance of a FL controlled rudder, even during tacking and jibing. A detailed description of the method including experimental validation is provided in Chapter 7.

Adaptive FL controllers have been presented as a promising approach to combine expert knowledge and new experiences automatically. In Velagic *et al.* [2003] a Sugeno type fuzzy inference system is combined with a feedback loop to adjust the scaling factors of the base fuzzy system.

For the aforementioned FL approaches expert human knowledge must be known a priori to design the fuzzy rule set. In contrast, Layne & Passino [1993] published a learning control algorithm which automatically generates the fuzzy controller's knowledge base on-line as new information on how to control the ship is gathered. Other examples of adaptive rudder control systems are based on artificial neural networks. Both Enab [1996] and Burns [1995] aim to provide an ANN-based system that can adapt its parameters towards optimal performance over a range of conditions without the need for manual adjustments.

An ambitious machine-learning approach to automatic rudder control was the *RoboSail* project (Van Aartrijk *et al.* [2002]). The project started in 1997 with the aim of building a self-learning autopilot for a single-handed sailing yacht. Agent technology, machine learning, data mining, and rule-based reasoning have been combined into a system which became commercially available after five years of development (Adriaans [2003]). After a few more years, in 2007 the Robosail Company stopped its business with the statement, that *"The market for adaptive autopilots was too small to sustain a healthy business in the long run."* (RoboSail [2011]).

A rudderless approach to automatic heading control of a sailing boat was presented by Benatar *et al.* [2009]. They have shown that control of a rudderless boat with two sails can be achieved by coordinating the two sails for the purposes of propulsion and turning. Figure 2.2 illustrates the basic principle.

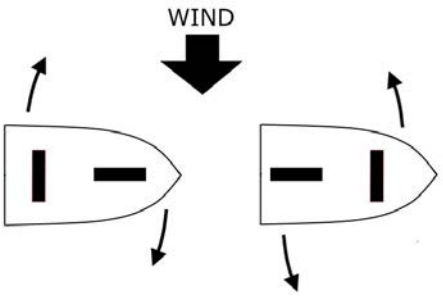


Figure 2.2: Turning the sail boat using sails only. The left example illustrates turning the boat towards downwind by increasing the resistance of the front sail, and decreasing that of the rear sail; the right example shows the opposite sail configuration which leads to the boat turning into the wind. (Benatar *et al.* [2009])

2.1.2 Automatic sail control

While extensive research has been carried out on automatic steering devices, automatic sail trim is a more recent idea and not yet well covered by scientific publications. Abril *et al.* [1997] identified the main reason for the lack of research on this topic as being the disuse of sails on merchant ships since the invention of the steam engine. Therefore the economic focus was clearly on motorized vessels. However, shortage of fuel is creating a rise in interest in alternative sources of energy. This and potential new applications in ocean observation bring sails back to discussion as an effective form of propulsion.

Rigging and sails

So far several different riggings[1] have been used on robotic sailing boats. They can be characterized according to the following criteria:

- Traditional fabric sail versus rigid wing sail
- Balanced versus unbalanced rig

[1]Rigging is the mechanical sailing apparatus attached to the hull in order to move the boat as a whole. This includes cordage, sails, and spars (masts and other solid objects sails are attached to)

In the history of sailing, which goes back several thousands of years, a large variety of different sail shapes and technologies have been used. Virtually all boats apart from those in recent sailing history used conventional fabric sails. This form of sails has some advantageous properties, especially when controlled by a human sailor. This includes the ease of reefing, repairing, and the fact that shape and camber can be altered by simply tensioning and releasing control lines.

By contrast, a wing sail is a rigid surface with an aerofoil cross-section similar to an aircraft wing. It can provide a much better lift-to-drag ratio than conventional sails (Shukla & Ghosh [2009]). Neal *et al.* [2009] highlight as a significant disadvantage of a wing sail that it is extremely difficult to design it in a way that it can be reefed reliably. Furthermore to construct strong, lightweight rotatable wings at reasonable cost is mentioned as an added difficulty. However, they maintain after extensive testing with different wing sails that the potential gains in reliability and efficiency would outweigh these problems.

Although most of the autonomous sailing boats featuring wing sails have been either designed for longevity (Neal *et al.* [2009]) or precision sailing (Elkaim [2006]) rather than performance, the America's Cup[1] 2010 was an impressive demonstration of the dynamic abilities of a rigid wing sail. The trimaran USA-17 (formerly known as BMW Oracle Racing 90 or BOR90) won the trophy with a rigid wing as its main sail.

On a conventional sloop rig, which is the most common rig type on sailing vessels, relatively high power is needed to tighten the sails against wind force. As being self-sufficient in terms of energy is one of the major goals in robotic sailing, the rig design has become the focus of attention. A balanced rig design (also known as Balestron rig, Aerorig^{TM}, swing rig, and EasyRig^{TM}) offers great potential in saving power (BalancedRig [2009]; Multirig [2009]). A balanced rig consists of an unstayed mast carrying a main[2] and jib[3] (see Fig. 2.3). The main boom extends forward of the mast (the mast passes through the boom) to the tack of the jib. The main and jib are sized in such a way that the force from the mainsail is slightly stronger than that from the jib. That is, the combined centre

[1]The America's Cup is a trophy awarded to the winner of the America's Cup match race between two yachts. The America's Cup is the oldest active trophy in international sport.

[2]A mainsail (also just main) is a sail located behind the main mast of a sailing vessel.

[3]A jib (also spelled jibb) is a sail set ahead of the mast of a sailing vessel.

of effort is just behind the mast. Therefore the load on the sheets is reduced by more than 50 % compared to a conventional rig due to the balanced distribution of the sail load caused by wind (Giger *et al.* [2009]).

Balanced rigs have been used on the autonomous sailing boats *Avalon* (Giger *et al.* [2009]) and *IBoat* (Briere [2008]). Furthermore, most of the rigid wing sails mentioned above can be considered to be balanced rigs.

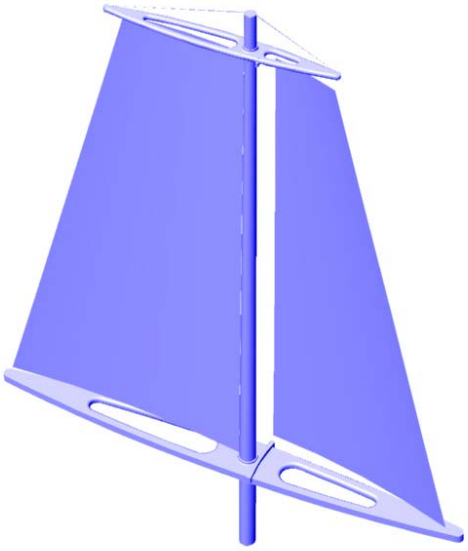


Figure 2.3: Balanced rig example (BalancedRig [2009])

Sail control strategies

The most basic control of the sail consists of setting its angle relative to the wind. Other aspects of sail trim, like reefing, altering of sail shape, or raking the mast go beyond the scope of autonomous sailing from a present-day perspective. On a conventional sailing boat sheets are used to control the sails. The sails are adjusted to create a smooth laminar flow over the sail surfaces. If the sheet is too loose, luffing[1] occurs to the sail. A common method for humans to adjust the sail is to pull the sheet in just so far as to make the luffing stop. This strategy cannot be easily applied to unmanned vessels, because both measuring the laminar flow along the sails as well as reliable detection of luffing is quite complicated.

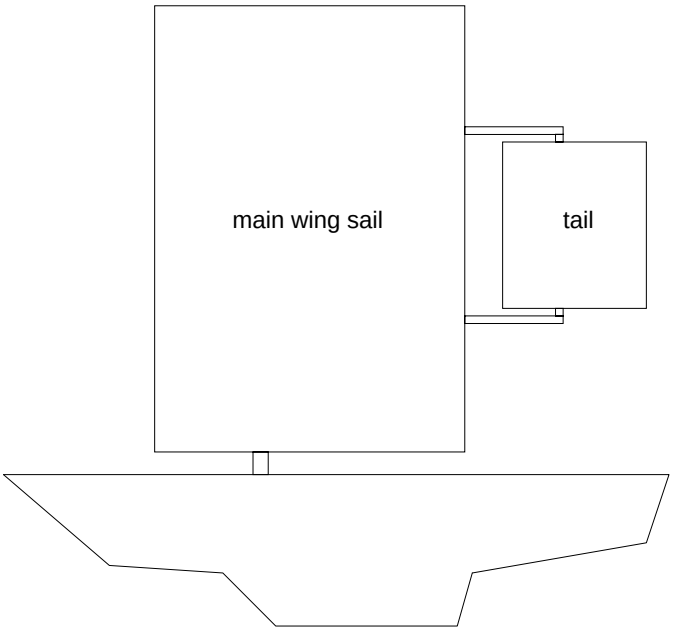
[1]It's called luffing when the sail flaps in the wind.

Most sail control strategies published for autonomous sailing boats rely on locally measured apparent wind data only (Abril *et al.* [1997]; Burnie [2010]; Giger *et al.* [2009]). While many of them have a virtually infinite number of sail positions (limited simply by the resolution of the actuator or the used data types) and therefore allow smooth sail control, just 10 discrete sail position are used on *MOOP* (University of Aberystwyth, UK) featuring a hysteresis condition to avoid continuous switching between two adjacent positions (Burnie [2010]). Reasons for a reduced number of sail positions are to save power on the sail actuator and to extend the lifetime of the sail gear. A state machine to allow for special sail trim during manoeuvres such as tack and jibe has been implemented on *Däumling* (University of Lübeck, Germany) *Avalon* (Swiss Federal Institute of Technology Zurich, Switzerland) and *IBoat* (ISAE, France) (Briere [2008]; Burnie [2010]; Giger *et al.* [2009]).

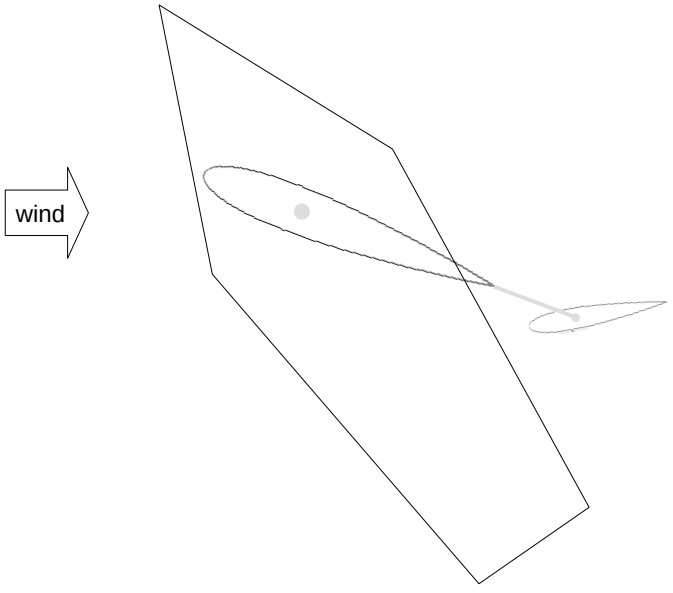
The author presents in Chapter 7 a novel method which does not directly calculate a sail position based on wind data. It firstly determines a desired heel[1] for the boat from the speed and direction of the apparent wind. A feedback-loop implemented as a Mamdani type fuzzy inference system (FIS) then controls the sail position towards this heel value.

All methods described above can basically be applied to both conventional and wing sails. For the latter a further control method has been presented in scientific literature, namely a self-trimming wing sail. A self-trimming arrangement typically consists of a wing sail vertically mounted on bearings that allow free rotation. A smaller wing called *tail* is usually mounted just behind the main wing (see Figure 2.4). An aircraft uses tails to control the exact angle of attack of its wings. Similarly, the tail on a wing sail system is able to control the thrust obtained from the wind and will automatically take into account any changes in wind direction (Worsley [2011]). Extensive research on self-trimming wing sails have been carried out by Elkaim & Boyce [2007]. Their experiments have shown upwind progress at $20-25$ deg and speeds of 60 % of the true wind speed under wind speeds of $12-25$ kn (approximately $6-13$ m/s) using a self-trimming wing sail on a 9.1 m catamaran.

[1]Heeling is the sidewards tilt of a sailing boat usually caused by lateral wind force.



(a)



(b)

Figure 2.4: Self-trimming wing sail: (a) side view of an arrangement with main wing sail and tail (b) orientation of wing sail and tail on a close hauled course

2.1.3 Ship routing

For motorised vehicles in isotropic, stationary environments, where a straight line is the shortest way to a target both in terms of distance and time, the identification of an optimum heading to reach the target is easy. This is significantly different for sailing boats, where a straight line route to the target may not even be navigable if the target is located upwind - the sailor has to beat to windward[1] in this case.

According to Spaans [1985] ship routing can be considered as the *"procedure where an optimum track is determined for a particular vessel on a particular run, based on expected weather, sea state and ocean currents"*.

Optimisation can be performed in terms of

- minimum passage time
- minimum fuel consumption
- safety of crew and ship
- best passenger comfort

or a combination of the criteria above (Motte *et al.* [1988]; Spaans [1985]). The present work focuses on minimum passage time. Fuel consumption is obsolete for exclusively wind propelled vehicles. Safety issues except for collision avoidance go beyond the focus of this study. Although most sailing robots are not intended to carry people, passenger comfort can partially be obtained by appropriate control of sails and rudder dependent on the boat dynamics.

Long term routing

The existing approaches for long term weather routing all require, more or less, certain weather predictions and a description of the boat's behaviour under certain wind conditions determined by experiment and typically formalised in a boat specific polar diagram (Spaans & Stoter [1995]; Thornton [1993]). The polar diagram describes the maximum speed a particular sailing boat can reach dependent on wind speed and direction.

[1] Beating to windward is referred to as the process of zigzagging when sailing upwind.

Most common computerised weather routing techniques are either an implementation of the manual isochrone plotting or optimisation methods within a discrete geographical grid system along the great circle route[1]. Motte & Calvert [1990] illustrated the effect of incorporating various discrete grid systems into a weather routing system, which employs Bellman's dynamic programming algorithm. Stawicki & Smierzchalski [2001] mentioned evolutionary algorithms as a promising approach to weather routing. Actual implementations of evolutionary path planning at sea have been published (Smierzchalski [2005]; Smierzchalski & Michalewicz [2000]) but do not address the special situation of sailing boats. All these approaches rely on weather forecast information on the one hand and sea charts on the other.

Philpott & Mason [2001] discuss two models to deal with uncertain weather data on ship routing. They consider the possibility of different weather conditions evolving in the future to determine routes which perform well under all of them. A recent implementations which has been tested on an autonomous sailing boat has been published by Giger *et al.* [2009]. They use on their boat *Avalon* a grid-based A* path planning algorithm based on weather forecasts mainly (A* search algorithm was introduced in Hart *et al.* [1968]).

Langbein *et al.* [2011] developed in a joint project with the author an algorithm for long term routing. It is the first implementation in the declarative rule-based programming language Constraint Handling Rules (CHR) (Frühwirth [2009]). It uses real-life wind forecasts which change over time, takes individual parameters of the sailing boat into account, and provides a graphical user interface. A more detailed description can be found in Appendix A.

Short course routing

The long term routing provides the boat with a series of waypoints which split the entire route into short legs. Aim of the short course routing is then to find an optimum way to the next waypoint given by long term routing. Due to the fact that short term weather is rather unpredictable short course routing deals with locally measured sensor data only. This is similar to a human sailor when

[1]Great circle route is the shortest route between two points on the surface of a sphere, e.g. the earth.

navigating to the next waypoint on a short regatta. These routes are usually optimised in terms of minimum passage time.

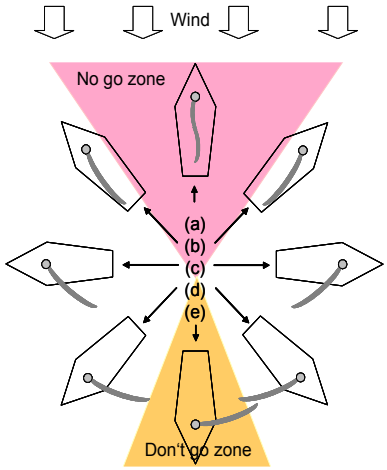


Figure 2.5: Points of Sail: (a) in irons (into the wind) (b) close hauled (c) beam reach (d) broad reach (e) running

The simplest short course routing strategy (ignoring obstacles) is to navigate in a straight line towards the next target if possible. If the straight line is not navigable (see *no go zone* in Figure 2.5) the boat sails a navigable heading pointing as closely towards the target as possible. This means sailing clause-hauled on an upwind course until the boat is back on a position where a directly navigable route is possible. Only a few more sophisticated approaches to short-course routing for autonomous sailing boats have been published so far: Philpott & Mason [2001] proposed a suitable model for short course racing. It treats the wind as a Markov process, and based on observations of the wind direction, it computes tacking and heading decisions at each point of the course so as to minimise the expected arrival time at the next mark. Giger *et al.* [2009] uses an A*-based routing algorithm for both long and short course routing.

The author presents a novel method for short course routing in Chapter 5. The calculation is based on the optimisation of the time-derivative of the distance between boat and target. It features a hysteresis condition which is of particular importance for beating to windward. The method shows its strengths especially when dealing with obstacles.

Collision avoidance

While basic sailing skills have already been applied to computer systems, the relevance of safety strategies, particularly collision avoidance becomes more and more important. Basically there exist two approaches in robotics how to deal with obstacles: (a) deliberative and (b) reactive methods.

Deliberative approaches use a model of the world by combining all relevant and available information. A route is calculated on the basis of this world model. If the world changes, the route has to be recalculated. These changes can either be altered obstacle information or changes in wind and weather data, which influence the world model. In a highly dynamic environment and a complex world model the computation power can be a serious limitation. By contrast, reactive methods rely just on locally measured sensor data and react spontaneously on it. No strategic decisions are made, or planning is done.

Deliberative approaches have their advantages especially in long distance navigation where available data (weather forecasts, topological data) are relatively stable. In contrast, reactive methods have their strengths particularly in reacting fast when in dangerous proximity to an obstacle.

Modern robot control architectures combine deliberative and reactive methods in order to make use of the advantages of both. Examples are Brook's subsumption architecture (Brooks [1986]) or Arkin's schema approach (Arkin [1992]). The author's layered architecture which is presented in Chapter 4 combines both approaches and focuses especially on the characteristics of sailing robots.

Collision avoidance in a maritime environment can be subdivided into two separate parts: (a) obstacle detection and (b) obstacle avoidance. The former involves techniques to detect and to classify potential obstacles on the water. Classification means to determine whether a detected object is an obstacle which has to be avoided or not. The latter describes actions to be taken on the basis of the result of obstacle detection.

Both obstacle detection and avoidance research have been carried out on autonomous underwater vehicles (AUV) (Showalter [2004]) and motorised ASVs (Benjamin *et al.* [2006]; Larson *et al.* [2007]; Lee *et al.* [2004]; Smierzchalski [2005]; Statheros *et al.* [2008]). Cameras, radar systems, and laser range scanners are the

most common sensors used so far. Obstacle detection methods from motorised vessels can be used on autonomous sailing boats as well.

By contrast, obstacle avoidance techniques from motorised ASVs cannot be used on sailing boats without adaptations. With the restrictions given by the physics of sailing it is not possible to navigate in any direction into the wind directly. Hence, it is not simple matter finding a reliable, fast and safe reactive obstacle avoidance strategy for a sailing robot. To the author's knowledge a raycast approach published by Sauze & Neal [2010] and the author's work (see Chapter 6) present the only relevant results in this field so far.

Sauzé and Neal proposed a simple method where the boat detects by raycasting on a raster based map when it is in close proximity to fixed obstacles. The method avoids headings which may put the boat in any immediate danger. Simulations have shown to find a safe route in most cases, however the boat sometimes became trapped in small inlets or between groups of tightly packed islands.

The author's method is an extension to his own short course routing strategy. It dynamically alters the underlying boat specific polar speed diagram by putting a penalty on directions where obstacles are located within a certain range. This results in a smooth obstacle avoidance behaviour. Details are given in Chapter 5.

2.2 Scientific community and events

2.2.1 Early examples

Prior to 2005 when the idea of *Microtransat Challenge*[1] initiated a new era of collaborative research in robotic sailing, a large number of autonomous underwater vehicles (AUV) had been developed (Blidberg [2001]; von Alt [2003]). However, research on autonomous surface vehicles (ASV), also known as autonomous surface crafts (ASC), was still in its early stage and mainly focused on motorised vessels (Caccia [2006]; Manley [2008]). Just a few researchers worked on fully autonomous sailing robots. According to their publications these teams seemed not to be well linked to each other. A few of the most noticeable early examples are described briefly here.

[1] `http://www.microtransat.org`

(a) (b) (c)

Figure 2.6: Early robotic sailing boats: (a) SKAMP – Station Keeping Autonomous Mobile Platform (b) RelationShip (c) Atlantis

Station Keeping Autonomous Mobile Platform (SKAMP)

The first attempt in autonomous sailing recorded in the literature is a project named *SKAMP (Station Keeping Autonomous Mobile Platform)*. The *SKAMP* was a wind propelled mobile surveillance platform which utilized a curving ring-shaped rigid wing sail (Figure 2.6(a)[1]). It was developed in 1968 by E. W. Schieben with the Radio Corporation of America and was optimised for autonomous station keeping rather than for dynamic performance (Schieben [1969]; Smith [1970]). Actual sailing data have never been published, so it remains unclear whether SKAMP ever sailed autonomously (Elkaim [2002]).

RelationShip

The second published autonomous sailing attempt was the *RelationShip* project of the University of Applied Science in Furtwangen, Germany (Figure 2.6(b)). The project started in 1995 with an ambitious plan to sail around the world

[1]Photo from Smith [1970]

with an unmanned trimaran. According to Elkaim [2002] the initial intention was to sail autonomously. However, after some difficulties the project changed to a remote control via satellite. After some years the project was cancelled due to regulatory difficulties. They did not get the permission to circumnavigate the globe with their unmanned RelationShip. The idea to declare the boat as flotsam did not convince the maritime authorities (Spiegel [1998]).

Fuzzy logic controlled sailing boat by Abril et al. [1997]

The first documented results of fully autonomous sailing have been published by Abril *et al.* [1997]. They presented a fuzzy logic controller for the rudder of a sailing boat. The desired sail position is a direct function of the apparent wind angle. Test runs have been carried out on a yacht model with an overall length of 1.03 m, a displacement of 4.5 kg and a sail area of 36.6 dm^2.

Atlantis

The *Atlantis* project of Stanford University began in 1997 with the concept of an unmanned, autonomous, GPS guided, wing-sail propelled sailing boat. The boat is based on a Prindle-19 Catamaran, with a self-trimming wing-sail (Figure 2.6(c)[1]). The maiden voyage took place in Redwood City Harbour in January 2001. (Elkaim [2002, 2006])

2.2.2 Competitions in robotic sailing

In many fields of robotics, competitions with memorable goals attract the attention of the media and the interested public, and can therefore provide a strong incentive for research and development in that particular area. The most popular examples in robotics are DARPA Grand Challenge for completely autonomous cars (Thrun *et al.* [2007]) or RoboCup soccer robots which aim to beat the human world champions by 2050 (RoboCup [2011]). The same happened to autonomous sailing during the first decade of the 21st century, when different events were organised almost at the same time.

[1]Photo from Loibner [1998]

Microtransat

Research into autonomous sailing has been recently stimulated by the *Microtransat* idea of Yves Briere (ISAE, France) and Mark Neal (Aberystwyth University, Wales, UK) (Briere [2008]; Briere *et al.* [2005]). The organizers describe the *Microtransat* on their web site as follows: *"The Microtransat Challenge is a transatlantic race of fully autonomous sailing boats. The race aims to stimulate the development of autonomous sailing boats through friendly competition."* (Microtransat [2011])

Participating sailing boats must be small (max. 4 m in length), unmanned, and use wind as the only form of propulsion. A few smaller Microtransat competitions took place prior to the real transatlantic race. These allowed contestants to exchange ideas and test their boats in less harsh environments.

The first Microtransat race was held on a lake in Saint Nicolas de la Grave near Toulouse, France, in June 2006. A total of three teams took part with their sailing robots and had to sail a course of about 2 km without human intervention. The author's first sailing robot prototype *Roboat I* was the only boat which completed the course successfully.

The second Microtransat competition was held in Aberystwyth, Wales, UK in September 2007. In order to come closer to the ultimate challenge of a fully autonomous crossing of the Atlantic Ocean this event was held on the sea in Cardigan Bay off the coast of Aberystwyth. A total of four boats participated, but just two of them took part in both races, a short race and a 24 h endurance race. The author's *ASV Roboat* was the only boat which demonstrated autonomous sailing over the full 24 hours.

The third Microtransat Challenge was the first attempt at crossing the Atlantic ocean. It started from Valentia, Ireland in September 2010. The only boat which entered the competition was *Pinta* from Aberystwyth University, Wales, UK. According to team member Colin Sauzé the boat was 49 h and 87 km under autonomous control before the computer system failed.[1]

[1] E-mail from Colin Sauzé on Microtransat mailing list from 02.10.2010

SailBot

Based on a successful student project at the University of British Columbia, Canada, where students worked on a robotic sailing boat, rules were developed for a friendly competition in autonomous sailing among university teams in North America. The first *SailBot* competition was held in 2006. The organisers describe *SailBot* on their web site as follows:

SailBot is an international competition for autonomously controlled sailboats. Aimed primarily at undergraduate student teams, the goal is to give engineering students a practical application of the topics they have learned, while also providing a fun way to learn project management in a multidisciplinary environment. A successful *SailBot* balances the needs of naval architecture, mechanical engineering, systems and electrical engineering, as well as project management.[1] *SailBot* competitions were held annually since 2006, except 2007.

SailBot is open for semi-autonomous and fully autonomous sailing boats up to 2 m in length. Since 2010 they provide an additional "open class" for boats with a maximum length of 4 m. For the 2011 competition five different events are announced: (1) fleet racing (2) station keeping (3) endurance contest (4) autonomous navigation and (5) presentation and design.

World Robotic Sailing Championship and International Robotic Sailing Conference

The World Robotic Sailing Championship (WRSC)[2] is an international competition for autonomous sailing boats. The first WRSC was held in Austria in 2008. Since then it has taken place every year: Portugal (2009), Canada (2010) and Germany (2011). The competition is open to boats of up to 4 m in length. The details of WRSC's rules change every year to respond to recent scientific developments and stimulate certain areas of research. By keeping a rather low entry threshold, the WRSC not only appeals to experts but also provides a platform for new teams in this recent field of research.

[1] Web site of SailBot 2011: `http://www.sname.org/SNAME/SailBot2011/Home/Default.aspx`

[2] `http://www.roboticsailing.org`

The competitions coincide with the International Robotic Sailing Conference (IRSC). This conference is the basis of scientific exchange in the robotic sailing community. The combination of competitions on the water and a scientific conference provides an opportunity to practically demonstrate theoretical developments.

2.2.3 Competing teams and their sailing robots

This section provides an overview of the teams which participated in recent robotic sailing competitions and are covered by scientific literature. Many of them have been encouraged by these events to start research in the field of autonomous sailing. Figure 2.7 shows the increasing number of boats competing in autonomous sailing competitions since their invention in 2006. The teams are presented here in alphabetical order.

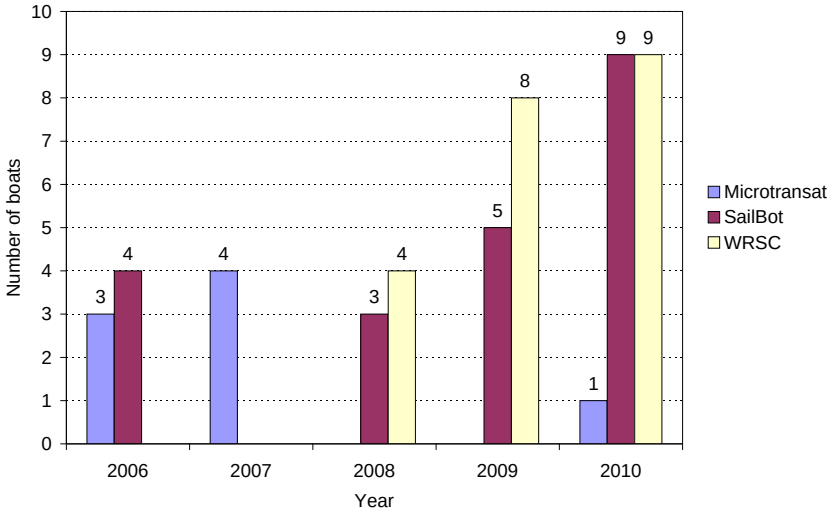


Figure 2.7: Number of boats competing in Microtransat, SailBot and WRSC. In 2010 SailBot and the WRSC were organised as a single event.

Austrian Society for Innovative Computer Sciences (INNOC) / De Montfort University (DMU)

The author formed his research team in robotic sailing within the framework of INNOC. This team focuses on control algorithms rather than on boat design.

(a)

(b)

(c)

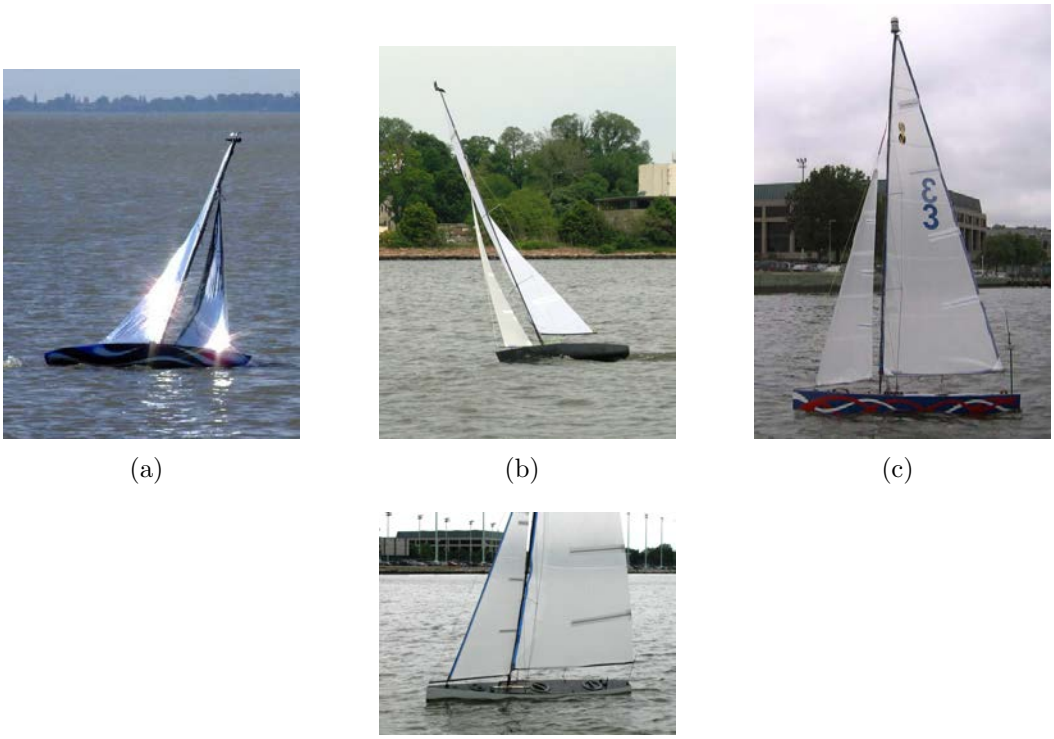
(d)

(e)

(f)

(g)

Figure 2.8: Autonomous sailing vessels with a length of less than 2 m: (a) Däumling, University of Lübeck (b) MOOP, University of Aberystwyth (c) Pi-mal-Daumen, University of Lübeck (d) Breizh Spirit, ENSTA Bretagne (e) Roboat I, INNOC (f) AROO, University of Aberystwyth (g) ARC, University of Aberystwyth



(a) (b) (c)

(d)

Figure 2.9: Autonomous sailing vessels with a length of exactly 2 m: (a) Black Adder, Queen's University (b) First Time, USNA (c) Gill the Boat, USNA (d) Luce Canon, USNA

(a) (b) (c)

(d) (e) (f)

Figure 2.10: Autonomous sailing vessels with a length of more than 2 m: (a) IBoat, ISAE (b) FASt, University of Porto (c) Pinta, University of Aberystwyth (d) Beagle-B, University of Aberystwyth (e) ASV Roboat, INNOC (f) Avalon, ETH Zurich

So far, two commercially available hulls have been adapted for the purpose of autonomous sailing.

Roboat I (Figure 2.8(e)[1]) is based on a ready-made yacht model of type *Robbe Atlantis* intended to be remote-controlled. It has an overall length of 1.38 m and weighs 17.5 kg. *Roboat I* won the Microtransat competition in 2006.

The basis for the *ASV Roboat* (Figure 2.10(e)[2]) is the commercially available boat type Laerling[3]. It has a length of 3.72 m and comprises a 60 kg keel-ballast and a total sail area of 4.5 m^2. *ASV Roboat* won the Microtransat 2007 as well as the WRSC in 2008, 2009, and 2010.

Further details about both robot sailing boat prototypes can be found in Chapter 3.

École nationale supérieure de techniques avancées (ENSTA) de Bretagne

A team of ENSTA Bretagne participated with their boat *Breizh Spirit* (Figure 2.8(d)[4]) in the WRSC 2009. Their design uses a custom built hull based on the IMOCA class design with a length of 1.3 m and two traditional sails. The control system is implemented on a PIC18F2550 microcontroller. (Sliwka *et al.* [2009])

Institut supérieur de l'aéronautique et de l'espace (ISAE)

IBoat from ISAE, France (Figure 2.10(a)[5]) is made of fibreglass and carbon and therefore is relatively lightweight. *IBoat* has a length of 2.4 m and a height of 3 m. It features 1.5 m^2 of sail area in a combination of two sails (main sail and jib) both mounted on a balanced rig. The sensors used are an electronic compass, a wind sensor (speed and direction) and a GPS receiver. The sensors are connected to a microcontroller via CAN bus (Briere [2008]). *IBoat* competed in the Microtransat 2006 and 2007 as well as in the WRSC 2009.

[1] ©INNOC
[2] ©INNOC
[3] `http://www.laerling.nl`
[4] ©Jan Sliwka
[5] ©INNOC

Queen's University

Mostly Autonomous Sailboat Team (MAST) was founded in 2004 at Queen's university. Their first vessel entering SailBot 2007 was *Black Adder* (Figure 2.9(a)[1]), a 2 m long carbon fibre hull with traditional sails using a PBasic Stamp for the control system. Since 2007 the team of undergraduate students have made significant modifications to their first design and participated in the Microtransat Challenge 2007, the WRSC 2008 and 2010 as well as SailBot 2008, 2009 and 2010. (Burnie [2010])

Swiss Federal Institute of Technology Zurich (ETH)

Avalon (Figure 2.10(f)[2]) was developed by a team of students from the Federal Institute of Technology, Zurich for the Microtransat challenge and participated in the WRSC 2009. *Avalon* features a monohull design with a length of 3.95 m, a balanced rig and a twin rudder system, and is powered by four solar panels of 90 W_p each, four lithium-manganese batteries of 600 Wh each and a direct-methanol fuel cell for back-up power. The control system is implemented on an MPC21[3] industrial computer running Linux. (Giger *et al.* [2009])

United States Coast Guard Academy (USCGA)

Intuition was developed by the United States Coast Guard Academy and participated in SailBot/WRSC 2010. The USCGA's 2 m monohull design features a conventional sloop rig with a sail area of 1.7 m^2. The control system is implemented on an ISIS PC104 single board computer running Windows XPe and MATLAB. (Burnie [2010])

United States Naval Academy (USNA)

USNA began their activities in 2007. They participated with their boat *First Time* (Figure 2.9(b)[4]) in SailBot 2008. In 2009 they entered *Luce Canon* (Fig-

[1] ©INNOC
[2] ©Patrick Moser
[3] `http://www.kontron.com`
[4] ©Paul Miller

ure 2.9(d)[1]) in SailBot and WRSC. In 2010 they participated with *Gill the Boat* (Figure 2.9(c)[2]) in SailBot/WRSC. The USNA team comprising undergraduate Naval Architects and Systems Engineers designs and builds new sailing vessels each year by continuously improving upon previous years' boats. Their designs use a custom built single hull with a length of 2 m and a conventional sloop rig with a sail area of about 3 m^2. (Miller *et al.* [2009, 2010])

University of Aberystwyth

The team working with Mark Neal and Colin Sauzé built multiple sailing robots varying in length between 50 cm and 3.5 m with the aim of performing long term autonomous missions for oceanographic monitoring.

AROO (Figure 2.8(f)[3]) was constructed in late 2004 as a proof of concept for a small but durable sailing robot. The hull is about 1.5 m long and is rigged with a 1 m high wing sail. (Neal [2006])

ARC (Figure 2.8(g)[4]) is about 1.5 m in length and features two independently controlled wing sails and two rudders controlled by a single actuator. It is equipped with a gimbaled compass, GPS receiver and a combination of an ATMega128 microcotroller and a Gumstix[5] single board computer running Linux. The only power source is a bank of 20 1.2 V AA size recharchable batteries with a capacity of 2500 mAh each. (Sauzé & Neal [2008])

Beagle-B (Figure 2.10(d)[6]) is their largest boat and was constructed in late 2006 by Robosoft (a French robotics company). It is 3.5 m^2 long and uses a 3 m solid wing sail. *Beagle-B* is intended to provide a serious oceanography platform for long term missions. Its power is provided by two 15 W solar panels and four 60 Ah 12 V batteries. It includes a YSI 660 Sonde for gathering oceanographic data as well as an Iridium SBD transceiver and GSM modem for data transmission. *Beagle-B* participated in the Microtransat Challenge 2007 in which it sailed a total of 25 km over 19 hours. (Sauzé & Neal [2008])

[1] ©Paul Miller
[2] ©Paul Miller
[3] ©Colin Sauzé
[4] ©INNOC
[5] `http://www.gumstix.com`
[6] ©Colin Sauzé

Pinta (Figure 2.10(c)[1]) was built for the WRSC 2008 and the Microtransat transatlantic race. Unlike the other boats in the team it uses a single traditional sail controlled by an electric winch system. *Pinta* is based on a Toper Taz sailing dinghy with a length of 2.95 m. The rudder is controlled by an off the shelf auto-helm. *Pinta* was the only boat to enter the 2010 Microtransat Challenge.

The *MOOP* (Mini Ocean Observation Platform; Figure 2.8(b)[2]) is a small lightweight sailing robot with an overall length of 0.72 m. Multiple *MOOPs* have been built so far with single or twin wing sail configuration, both with rudders and rudderless. They are controlled either solely by a PIC microcontroller or with a combination of a PIC and a Gumstix Single Board Computer. Several *MOOPs* took part in the SailBot and WRSC competitions in 2009 and 2010. (Burnie [2010])

University of Lübeck

The University of Lübeck started their autonomous sailing activities in 2009. They participated in Sailbot/WRSC 2010 with two boats.

Däumling (Figure 2.8(a)[3]) is based on a University Club (Graupner, Germany) boat model with a length of 0.53 m and two sails with a total area of 0.145 m^2. The boat is inteded to be used as a testbed for various methods of artificial intelligence. (Bruder *et al.* [2009])

Pi-mal-Daumen (Figure 2.8(c)[4]) is based on a standard IOM (International One Meter class) hull with a length of 1 m and two sails with a total sail area of 0.4 m^2. On board control is provided by a custom built circuit board featuring an ATMega2560 microprocessor. (Ammann *et al.* [2010])

University of Porto

Team *FASt* (Figure 2.10(b)[5]) developed a 2.5 m long custom built hull and has a total sail area of 3.7 m^2. The design was inspired by modern ocean racing

[1] ©Colin Sauzé
[2] ©Colin Sauzé
[3] ©Alexander Schlaefer
[4] ©Alexander Schlaefer
[5] ©José Carlos Alves

yachts. The control system is implemented on a small FPGA-based single board computer, including a 32-bit RISC microprocessor running at 50 MHz. Communication with the boat is possible using Wi-Fi, GSM, Iridium SBD and a conventional RC receiver used in radio-controlled models. *FASt* was entered in the WRSC 2008 and 2009. (Alves *et al.* [2008])

2.3 Summary

While completely autonomous sailing boats are a more recent idea, extensive research has been undertaken on semi-autonomous systems, where just a subset of the functionality is covered. The first task to be automated was the governing of the rudder. In the early 20^{th} century steering engines became more common on ships and a number of automatic steering systems were subsequently invented. By now all big manufacturers of marine electronics offer electronic self-steering systems which keep a boat on a predefined compass course or a heading relative to the wind direction.

Automatic sail trim has not been studied very well so far. Most sail control strategies published for autonomous sailing boats rely on locally measured apparent wind data only. By contrast, the author presents in Chapter 7 a fuzzy logic control strategy which does not directly calculate a sail position based on wind data. So far several different riggings have been used on robotic sailing boats. Because of limited power and longevity requirements on robotic sailing boats balanced rigs and rigid wing sails become an alternative to traditional sloop rigs.

Another research topic is route optimisation for sailing robots. Most common computerised long term routing techniques are either an implementation of the manual isochrone plotting or optimisation methods within a discrete geographical grid system along the great circle route. For short course routeing (based on locally measured wind data only) the most common and simplest method is to sail a straight line towards the target if possible. If not, the boat sails clause-hauled until a directly navigable route is possible. Only a few more sophisticated approaches to short-course routing for autonomous sailing boats have been published so far. One of them is presented in detail in Chapter 5.

A further important problem to be solved for long-term unmanned and autonomous missions on the sea is reliable obstacle avoidance. While collision avoidance research has been carried out on autonomous underwater vehicles (AUV) and motorised ASVs, there has not been taken place much research on obstacle avoidance for robotic sailing boats so far. The author presents a novel reactive method for this purpose in Chapter 6.

Research into autonomous sailing has been recently stimulated by the robotic sailing competitions Microtransat, SailBot, and World Robotic Sailing Championship (WRSC). Microtransat and SailBot started in 2006, WRSC in 2008. While just a few researchers worked on fully autonomous sailing robots before, the number of research groups in robotic sailing is growing rapidly since 2006. Many novelties in robotic sailing have been published on the International Robotic Sailing Conference (IRSC). This conference was founded by the author and annually held since 2008.

Chapter 3

Research methodology and experimental setup

Similar to all relevant research groups in robotic sailing (see Section 2.2), the author used robotic sailing boat prototypes for field experiments as one of his main research method. Two robotic sailing boats have been developed especially for this purpose. Furthermore simulations were useful to evaluate the novel algorithms for short course routing and obstacle avoidance.

The author started with a 1.38 m long first prototype *Roboat I*, based on an off the shelf yacht model (see Section 3.1 for details). The advantages of such a small robot are that it is cheap, lightweight, easy to handle and easy to build. Test runs can easily be arranged without the need for any special infrastructure (slip ramp, crane) or a chasing boat. On the other hand a boat of this size is extremely sensitive even to small waves and wind gusts. This makes it difficult to reproduce experimental results and to evaluate the implications of minor changes in the control system. Furthermore, with its restricted space for additional equipment and a relatively short operating time, it is not a serious platform for maritime applications.

Due to these limitations a second and significantly larger prototype named *ASV Roboat* has been built (see Section 3.2 for details). This boat is 3.72 m in length and provides enough space for additional equipment, which enables it to be used for the first real-world applications of autonomous sailing technology.

Furthermore, due to the fact that both boats run the same software, the scalability of the control methods can be demonstrated.

In addition to tests on the water, some scenarios have been evaluated in a computer simulation. A simple software model of the boat's behaviour allowed for evaluation of routing and obstacle avoidance methods on the computer.

The approach taken does allow for appropriate comparison between the performance of the boats and simulation. Other evidence is provided by winning of robotic sailing competitions (see Section 1.6).

Table 3.1 provides an overview of which research tools were used with regard to the particular research topics.

Research topics	Roboat I	ASV Roboat	Simulation
Sailing boat routing	✓	✓	✓
Obstacle avoidance			✓
Sail and rudder control	✓	✓	
Communication		✓	
Control architecture	✓	✓	

Table 3.1: Overview of tools used to address the research topics

3.1 First prototype: Roboat I

Roboat I (see Fig. 3.1) is the first of two robot sailing boats that were designed as part of this study. The aim of these prototypes was to test out in practice the algorithms that had been developed. Since the goal of the study is to design processes that can be applied as far as possible to all standard types of sailing boat, the boat was not constructed from new; rather, a commercially available model boat was used as the starting point.

Roboat I was converted into a robotic sailing boat at the beginning of 2006 and was used for experiments in the context of this study up to the middle of 2007. It was then replaced by the ASV Roboat (see Section 3.2) as the research platform.

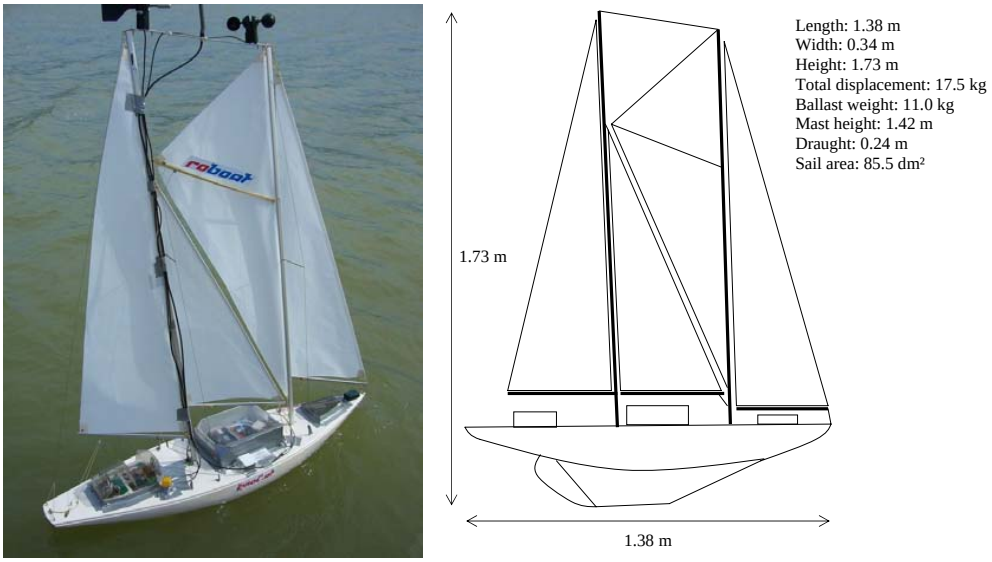


Figure 3.1: Roboat I

3.1.1 The Robbe Atlantis model remote-controlled sailing boat

In various internet forums devoted to model sailing boats, the Atlantis model (manufactured by Robbe) was repeatedly described as very easy to sail, robust and usable in comparatively strong wind for model sailing boats.

It is a gaff-rigged schooner[1] with a length of 1.38 m, a height of 1.73 m and a total displacement of 17.5 kg. The rig contains four sails which comprise a total area of 85.5 dm^2.

All four sails are controlled by just one winch servo. The second actuator is also a standard model servo and controls the rudder. The boat is controlled by a standard radio remote control with at least two channels (rudder, sails). In our case, a Robbe Futaba F14 was used at a frequency of 40 MHz.

In addition, the boat is fitted with an auxiliary electric drive that makes it possible to carry out a range of tests independently of the wind, especially those relating to the routing algorithm, and to bring the boat safely back to shore in dead calm.

[1] A schooner is a type of sailing vessel characterized by the use of two or more masts with the forward mast being no taller than the rear masts.

The transition from a radio-controlled sailing boat to a fully autonomous sailing boat required the addition of a few sensors and some computing power. In addition, the boat was fitted with a communications infrastructure to make it possible to receive data from the boat on land in real time and to be able to adjust the configuration settings.

3.1.2 Computer and communications

PIC microcontrollers from Microchip[1] were used to prepare the sensor readings and to control the actuators through the hardware. These microcontrollers transmitted the readings using the specially designed *Simple Sensor Network* protocol (SSN, see Appendix C) to the main computer, which in turn is responsible for implementing all of the sailing logic.

A VIA EPIA-MII 6000E Mini-ITX PC with 600 MHz, 512 MBRAM,2 GB SD card running Debian Linux was used as the central computer. The microcontrollers were connected via RS232 interfaces. Communication with the land was carried out via a PCMCIA WLAN card, which was connected to an access point on shore. Despite the relatively high demands in terms of power and space, it was decided to opt for this computer platform for the following reasons:

- high computing power
- support for a wide range of programming languages
- wireless LAN connection
- sufficient storage space for extensive logging
- extensive debugging options

The signals that the main computer sends to the actuators via the microcontrollers can be overridden by remote control. If the need arises, it is therefore possible to intervene and steer the boat by remote control.

[1] `www.microchip.com`

3.1.3 Sensors and actuators

The following sensors were fitted to the boat:

- wind speed: paddle wheel with reed contact switches
- wind direction (apparent wind): wind vane with integrated potentiometer
- position and speed (above ground): GPS receiver with serial NMEA0183 interface
- heading and tilt: PNI TCM 2.5 tilt-compensated 3-axis compass module

Because of the low resolution of the on-board wind direction sensor and the relatively low speed of the boat, it was not possible to calculate a sufficiently precise value for the true wind direction from the apparent wind, heading and speed. For the purpose of route planning, the true wind was therefore also measured on land and sent to the boat by WiFi.

The actuators for the rudder and sails are standard model-making servos and were adopted without modifications from the kit that came with the remote-control sailing boat.

3.1.4 Power supply

The power supply for the boat was provided by six 7.2 V racing packs of 1800 mAh each. At an average power consumption of approx. 20 W, the boat can sail continuously for about 4 h. The main consumer is the Linux computer at approximately 15 W.

3.2 Second prototype: ASV Roboat

The ASV Roboat is the second prototype used for this study and it was built to take part in the Microtransat Challenge 2007 in the Irish Sea off the coast of Wales, UK. The Roboat I would not have been suitable for the conditions that prevail there. The 24-hour competition, in particular, made a larger boat essential to accommodate the on-board batteries required.

Furthermore, it was hoped that the larger boat would allow the experiments to be reproduced more accurately, as small waves or gusts of wind do not have such a large influence on handling as was the case with the significantly smaller Roboat I.

3.2.1 Laerling class boats

The basis for the ASV Roboat (Fig. 3.2) is the commercially available boat type Laerling. The boat was originally created for kids to learn sailing, and therefore safety and stability are its major characteristics. It has a length of 3.75 m and comprises a 60 kg keel-ballast, which will bring the boat upright even from the most severe heeling. Including batteries the overall weight of the boat is about 300 kg. The sail area of mainsail and foresail together is 4.5 m^2.

Figure 3.2: ASV Roboat at field tests on the Baltic Sea (2011)

Fig. 3.3 shows the technical infrastructure that was set up during the conversion to a robotic sailing boat. It includes sensors, actuators and power supply. The individual areas will be discussed in more detail below.

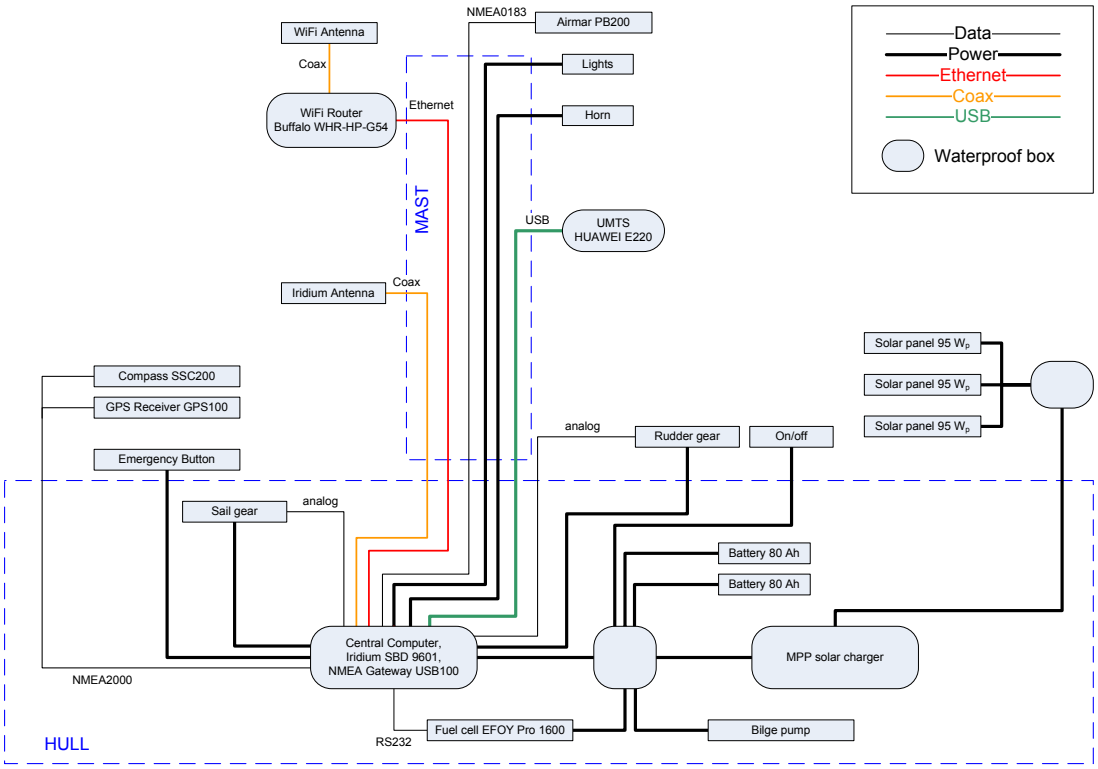


Figure 3.3: Technical infrastructure on ASV Roboat

3.2.2 Computer and communications

The control software runs on a Linux-based on-board computer system using incoming data from various sensors (GPS, compass, anemometer, etc.).

Exactly the same computer is used as for the first prototype, the Roboat I (see Section 3.1). The computer, together with the other electronic components, is housed in a water-tight case (Peli 1500, Protection Class IP67) and connected with the sensors and actuators on the boat using plugs of Protection Class IP67 (Phoenix M12 and Phoenix 7/8"series) (Fig. 3.4).

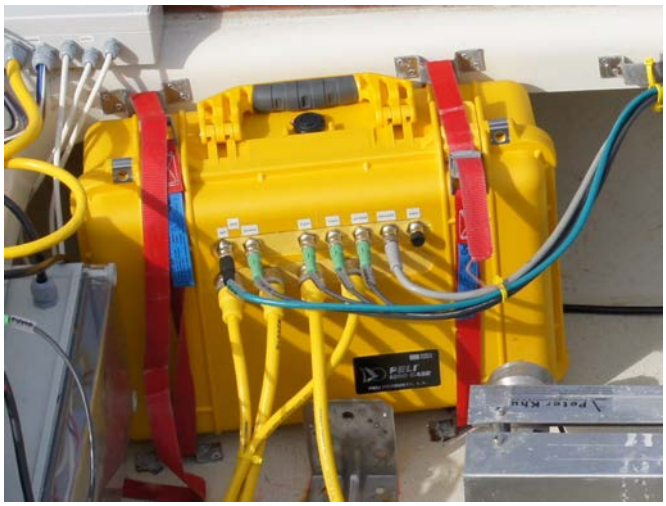


Figure 3.4: Waterproof box with IP67 Connectors

The *ASV Roboat* features a three-stage communication system, combining WLAN, 3G and an iridium satellite communication system[1], allowing continuous real-time access from shore. The detailed set-up is described in Chapter 4.2.

Although the boat fundamentally sails itself, in some cases it may be necessary to take control using the actuators. To make this possible, remote control software was developed that runs on a netbook with a touchscreen (Fig. 3.5). The software is written in Java and communicates with the boat via WiFi.

[1] `www.iridium.com`

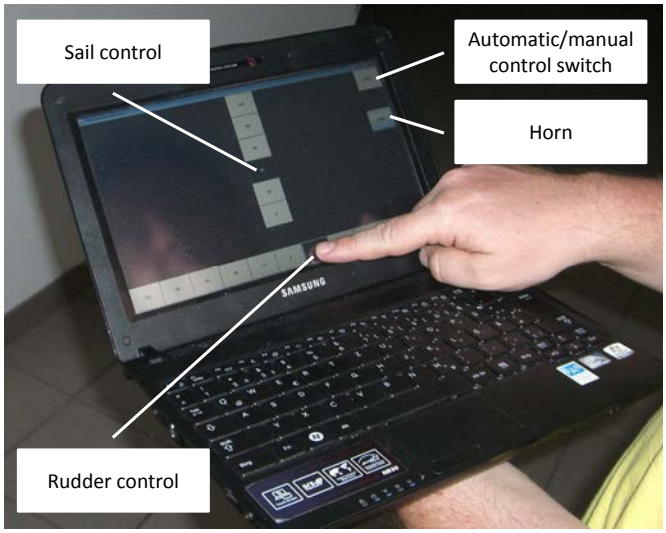


Figure 3.5: Remote control software written in Java running on a netbook with touch screen

3.2.3 Sensors

The following environmental data must be measured by sensors to control the ASV Roboat:

- wind speed
- wind direction
- heading
- rate of turn
- heel angle
- boat position

In addition, other values are measured that are not directly required for navigation, but are recorded every second:

- boat speed
- depth of water

- water temperature
- air temperature
- humidity
- atmospheric pressure

In order to increase the availability of the system, many of the sensors are redundant. If a sensor fails, a secondary sensor can be activated.

The PB200 digital weather station manufactured by Airmar is used as the primary sensor unit. This device contains all of the sensors required to take the environmental readings. Table 3.2 summarises the information that the Airmar PB200 provides to control the ASV Roboat. The data is transferred in the NMEA0183 serial format.

NMEA0183 sentence	**Description**	**Frequency**
$WIMWV	Wind speed, apparent wind direction	2 Hz
$HCHDT	Heading	2 Hz
$TIROT	Rate of turn	2 Hz
$YXXDR	Pitch, roll	1 Hz
$GPRMC	GPS data (position, speed)	1 Hz

Table 3.2: Primary sensor data on ASV Roboat from Airmar PB200 (Airmar [2009])

Table 3.3 provides a summary of the secondary sensors that are connected to an NMEA2000 bus, which is also fitted on board the boat. This is connected to the computer via a gateway (Maretron USB100). This gateway in turn converts the NMEA2000 records into NMEA0183 records, which can then be read via the serial interface in the computer.

3.2.4 Actuators

In order to sail a boat autonomously, the rudder and sails must be adjusted automatically. Electric drives were designed for both and mounted on the ASV Roboat.

Device	NMEA0183 sentence	Description	Frequency
Maretron SSC200	$IIHDG	Heading	10 Hz
	$IIROT	Rate of turn	5 Hz
	$PMAROUT	Pitch, roll	1 Hz
Maretron GPS100	$IIGGA	GPS data (position, speed)	1 Hz

Table 3.3: Secondary sensor data on ASV Roboat from Maretron NMEA2000 bus (Maretron [2005, 2006])

Initially, the original rudder of the ASV Roboat was used. A linear drive was connected to the tiller to control the rudder (Fig. 3.6). However, this rudder drive was replaced in 2009 with a balanced rudder system specially built for the ASV Roboat to reduce power consumption (Fig. 3.7).

Figure 3.6: Linear actuator connected to the tiller controls rudder (2006–2008)

Fig. 3.8 illustrates the concept of a balanced rudder. The rotational axis of the rudder is shifted to a point approximately in the middle of the rudder blade. This means that when the rudder is turned, the flow of water actively works on the forward part to increase the angle of deflection, whereas the same flow acts on the rear part to reduce the angle. The area in front of the axis is kept slightly less than that behind in order to avoid rudder instability.

The ASV Roboat has two sails (mainsail and foresail). The two sails are controlled by a single drive; in other words, they are always trimmed in parallel. The

Figure 3.7: Self-constructed rudder gear with balanced rudder (since 2008)

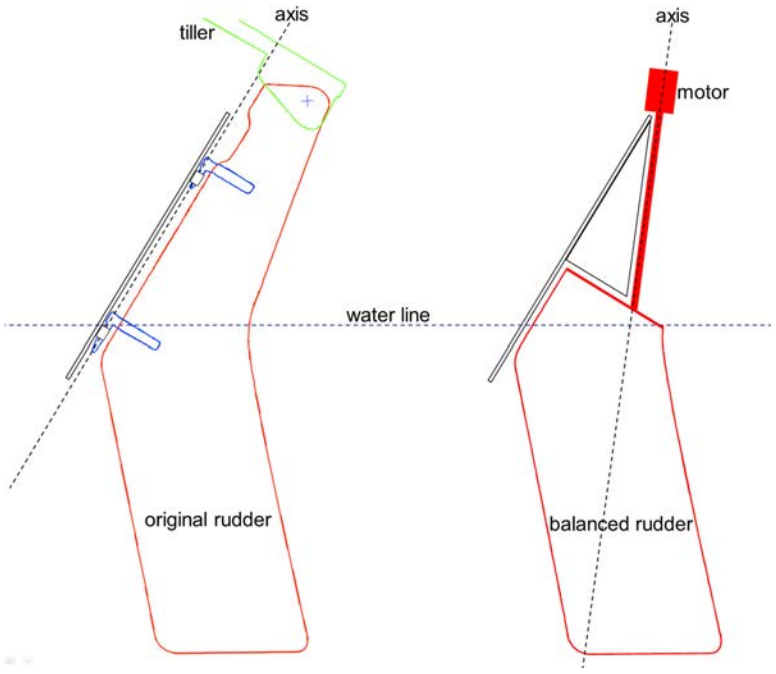


Figure 3.8: Illustration of concept of balanced rudder

mainsheet is operated with a 2:1 gear reduction, while the jib sheet is attached directly to the drive. The mainsheet therefore covers twice the distance of the jib sheet. (see Fig. 3.9)

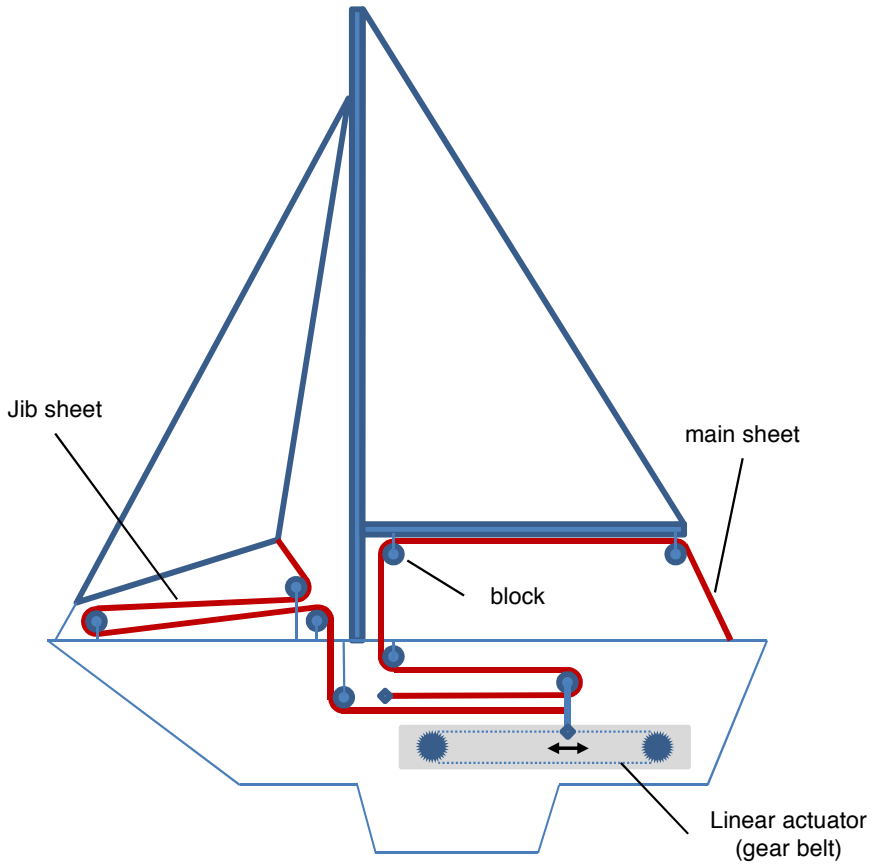


Figure 3.9: Sheet guidance and sail gear on ASV Roboat

Sailing ships normally use winches to control the sheets. As the sheets sometimes become slightly entangled as a result, a linear drive has been designed for the ASV Roboat instead. The drive consists of a slide that is moved by a belt. The maximum travel of the slide is 75 cm. The jib sheet can therefore be tightened and slackened by 75 cm and the mainsheet by 150 cm. A self-locking worm gear on the motor ensures that power is only consumed when the sail position is being adjusted.

In order to make the adjustment of the fore sail easier, especially during tacking, the ASV Roboat was fitted with a self-tacking jib (see Fig. 3.10). A

self-tacking jib is a fore sail that does not require operation of the jib sheet when tacking, as it automatically changes from one side to the other. The jib sheet does not lead directly to the sail drive but is diverted over rollers. The first roller moves freely on a curved rail mounted in front of the mast. From there, the jib sheet is fed forward to a deflection roller near the bow, before running back into the cockpit, where it is attached to the sail drive.

Figure 3.10: Self-tacking jib on ASV Roboat

In addition to the rudder and the sails, there are other actuators on board, but these are not directly required for autonomous navigation:

- navigation light
- fog horn
- bilge pump (with automatic switch)

3.2.5 Power supply

The average power consumption of the ASV Roboat is approximately 35 W. This measurement was taken during a test sailing on the Baltic Sea at an average wind speed of 6.5 m/s.

The main power source of the ASV Roboat are solar panels providing up to 285 W_p of power during conditions of full sun. 285 W_p corresponds to approximately 30 W of average output over a whole year under central European weather conditions (Finanztest [2006]; Schrag [2011]). To cover the night period, electricity is stored in two deep-cycle batteries of 1.92 kWh together (80 Ah at 12 V each).

In order to be able to operate the boat at least for a limited period in conditions of reduced sunlight (night, bad weather) or if the solar power system breaks down, the boat is also equipped with a direct methanol fuel cell (EFOY Pro 1600). It delivers 65 W and features as a backup energy supply. The fuel tank contains 28 l of methanol. With a methanol consumption of 1.11 l/kWh as stated in the data sheet, the boat can operate about a month with the fuel cell as its only source of electricity.

The boat is therefore currently showing a slightly negative energy balance (see Table 3.4). The largest consumers are the central computer, the WiFi access point and the drive for the sail trim. These three consumers together account for approximately two-thirds of the power consumption.

The following steps can be taken to lower the power consumption and achieve a positive energy balance (see Table 3.5):

- *Rig design:* a balanced rig (also known as Balestron rig, Aerorig^{TM}, swing rig and EasyRig^{TM}) provides great potential to save power (BalancedRig [2009]; Multirig [2009]). According to Giger *et al.* [2009] and based on experience with the balanced rudder system of the ASV Roboat (see Section 3.2.4), we anticipate a potential saving of at least 50 per cent from a balanced rig design.

- *Computer migration:* by converting to an energy-efficient embedded industrial PC, the power consumption can be reduced further. A Vortex86SX 300 MHz could be used, for example. This computer runs Linux and provides all of the interfaces required. Its power consumption of 3.24 W is almost 80 per cent lower.

- *Avoiding short-range communication:* WiFi and 3G communication are extremely practical when it comes to being able to receive large amounts of

data from the boat and adjusting the settings of the boat during accompanied trial runs. However, in most applications the boat operates unmanned, unaccompanied and outside the range of these two technologies. In these cases, WLAN and 3G are not practical and the WLAN access point and 3G modem on board can be switched off. The iridium satellite modem can remain on standby to enable communication with the boat if required.

- *Removing sensors:* currently almost all of the sensors are present on board in duplicate, once in the Airmar PB200 using NMEA0183 and secondly on the NMEA2000 bus. The entire NMEA2000 bus could therefore be removed without losing any significant sensor data for autonomous sailing. Of course, availability could suffer if redundant systems are omitted.

- *Increase in efficiency in the control algorithm for rudder and sails:* methods that require little adjustment of the rudder and sail settings can lead to significant power savings. However, this always involves a compromise between sailing quality and energy consumption. The specific application for which the robot sailing boat is being used must always be considered when it comes to making adjustments of this sort. Power efficiency in the control algorithms has not been considered in this study and therefore provides room for further research.

Component	Model	Power	Pct.
Embedded PC	VIA EPIA-MIII	15.20 W	44.0 %
Power supply	ATX	0.31 W	0.9 %
GPS receiver	Maretron GPS100	1.80 W	5.2 %
Compass	Maretron SSC200	1.62 W	4.7 %
Depth, speed, water temperature sensor	Maretron DST100	0.58 W	1.7 %
NMEA2000 protocol converter	Maretron USB100	1.82 W	5.3 %
Weather station	Airmar PB200	2.64 W	7.6 %
WiFi access point	Buffalo WHR-HP-G54	3.4 W	9.8 %
3G modem	HUAWEI E220	2.5 W	7.2 %
Satellite modem	Iridium SBD 9601 (switched on 2 min/h)	0.06 W	0.2 %
Sail gear	self-construction (non-balanced sloop rig)	4.58 W	13.3 %
Rudder gear	self-construction (balanced rudder)	0.02 W	0.1 %
Overall sum	**34.53 W**		

Table 3.4: Power consumption of current ASV Roboat (September 2011)

Component	Model	Power	Pct.
Embedded PC	Vortex86SX	3.24 W	39.3 %
Weather station	Airmar PB200	2.64 W	32.0 %
Satellite modem	Iridium SBD 9601 (switched 2 min/h)	0.06 W	0.7 %
Sail gear	self-construction (balanced rig)	2.29 W	27.7 %
Rudder gear	self-construction (balanced rudder)	0.02 W	0.2 %
Overall sum	**8.25 W**		

Table 3.5: Power consumption: optimised configuration shows great potential for saving electric energy.

3.3 Summary

Two fully autonomous sailing robots have been developed as a basis for real-world experiments. After first tests with the 1.38 m long prototype *Roboat I* the significantly larger *ASV Roboat* has been built. This boat is 3.72 m in length and provides enough space for additional equipment, which enables it to be used for the first real-world applications of autonomous sailing technology.

The succeeding Chapter 4 describes the software and communication architecture which is a flexible and reliable platform to run software modules for all parts of autonomous navigation.

Chapter 4

System architecture

Both sailing robot prototypes *Roboat I* and *ASV Roboat* (see Chapter 3 for details) use the same system architecture which consists of two major parts:

- **Four layered hybrid subsumption architecture:** Mobile autonomous robot systems are usually divided into separate layers each responsible for a part of the problem. Brooks [1986] presented the subsumption architecture, which has been widely influential in autonomous robotics. Das *et al.* [2011] describe subsumption as *"a way of decomposing complicated intelligent behaviour into many simple behaviour modules, which are in turn organized into layers"*. They further state that *"each layer implements a particular goal of the agent, and higher layers are increasingly abstract"*.

  Basically two different approaches exist to realise autonomous behaviour: deliberative and reactive. In deliberative systems sensor data together with a priori knowledge about the environment is used to generate a model of the world in which planning occurs. Planning mechanisms generate a detailed plan for the robot's actions to reach a previously specified goal. After the plan is completed the robot will act according to it. These systems have shown good performance in complex static environments, however generating a plan is usually a time intensive task. Thus, these systems cannot react too quickly in dynamically changing and unpredictable environments. The reactive approach connects measured sensor data directly with the robot's actuators. Therefore, the robot can respond rapidly to changes in the world

like unexpected and moving obstacles. Reactive or behaviour-based robots have shown great performance in constantly changing environments often found in real world tasks. Since the robot only acts on local information without global knowledge about the environment it may not reach a global optimum and lack the ability to perform complex tasks.

In order to exploit the advantages of both reactive and deliberative approaches in a modular system, a four layer hybrid subsumption architecture has been developed as a framework for autonomous sailing boat navigation methods.

- **Three stage communication system:** During the development process of an autonomous sailing boat it is very important to have a permanent data link to the vessel which is capable on the one hand of taking over control manually for safety reasons if the artificial sailor on board does not work as expected, or an obstacle crosses the boat's trajectory and makes manual intervention necessary. On the other hand it is quite convenient to have an opportunity for real-time monitoring from shore in order to identify room for improvement in the control algorithms of the boat.

  Not only for testing and optimisation purposes, but also for most of the possible applications of autonomous sailing boats a reliable communication system between shore and boat is essential in order to transmit collected data. At least geographic coordinates of the boat or of the next target need to be transmitted for the applications mentioned earlier in Section 1.3.

The control architecture and the communication system proposed in this chapter represent a flexible and robust basis for implementing and testing of novel methods in autonomous sailing (see Chapters 5–7).

4.1 Layered architecture

In order to control an autonomous sailing boat, data about the environmental conditions are necessary. The system calculates a desired position for rudder and

sails based on sensor information. These are the only actuators needed to steer a sailing boat autonomously.

Sensors deliver real-time information about current wind direction and wind speed. Additionally, heeling (transverse inclination of a sailing boat), geographic position, and heading are measured on the boat. These are the minimum data needed for the autonomous sailing boat used in the experiments for short course routing (Chapter 5) and manoeuvre execution (Chapter 7). For obstacle avoidance (Chapter 6) additional information about shape and position of obstacles is extracted from nautical charts. Furthermore, weather forecasts can be taken into account for long term routing. Long term routeing goes beyond the scope of this thesis but has been treated in a joint project of the author with University of Ulm, Germany (see Appendix A for details).

Although the sailing boat is designed to operate completely autonomously, a human operator has to predefine strategic goals. These prerequisites include the end point of the sailing trip and optional intermediate way points to be passed, such as buoys of a regatta or ports.

The actual control system is divided into four layers. Each layer has access to sensor data by connecting to the data abstractor. The abstractor is a computer program which is executed directly on board of the boat. It gathers sensor data and transforms the raw data into semantically usable values. Preprocessing like damping, scaling, unit transformations, or plausibility checks are all done at this level. The four layers represent separate parts of autonomous sailing boat navigation:

- Strategic long term routing
- Short course routing
- Manoeuvre execution
- Emergency reflexes

In addition to sensor data each layer gets prerequisites from the preceding super ordinate layer. The task of each layer is to satisfy the prerequisites as accurately as possible (Figure 4.1). A more detailed description of the tasks of each layer is given below.

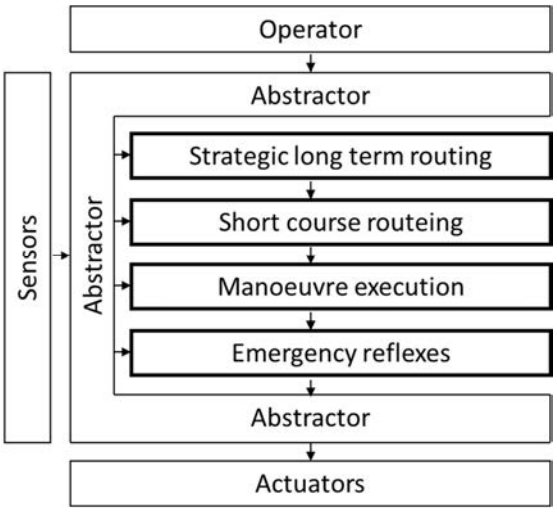


Figure 4.1: Four layered hybrid subsumption architecture for robotic sailing

4.1.1 Strategic long term routing

The routing algorithm determines an optimal rough route with respect to the boat-specific behaviour, the predicted weather conditions and sea topology. The route is divided into many short legs and described as an ordered set of coordinates to be passed. The next target coordinate is handed on to the layer below, the short course routing layer.

There are various commercial applications for long term weather routing which can be used at this point (e.g. Visual Passage Planner 2[1], SailFast[2]). Almost every commercial application uses the so-called isochrone method for calculation of routes. A novel approach which has been developed by the author in cooperation with Johannes Langbein und Thom Frühwirth (Ulm University, Germany) is presented in Appendix A.

4.1.2 Short course routing

In order to steer a sailing boat towards a specific target, a navigable route has to be specified in advance. Not all points of sail are navigable ("No go zone" in Figure 2.5). Points of sail is the term used to describe a sailing boat's course in

[1] `www.digwave.com/products.htm`
[2] `www.sailfastllc.com`

relation to the wind direction. Some courses are navigable, but quite inefficient ("Don't go zone" in Figure 2.5). These restrictions have to be taken into account in short course routing. Therefore the route may contain multiple sections, connected by manoeuvres such as tacking or jibing. Also change of wind direction while sailing a stable compass course may cause a manoeuvre.

The aim of the short course routing layer is to find an optimum way to the next target which is given by the strategic long term routing layer. As the local wind conditions often change and accurate weather forecasts are not available for very short distances and periods, only local and present wind conditions are taken into account in order to determine an optimum heading for the vessel. The short course routing has to provide an answer to the perennial question when to tack on upwind courses, which cannot be directly sailed. The short course routing layer does not need a weather forecast at all. It reacts to changes in wind conditions in real-time by recalculating the optimum heading periodically. Fixed obstacles like islands or shoals as well as moving obstacles like other vessels have to be considered here. This optimum heading acts as input for the manoeuvre execution layer.

Details about the novel method for short course routing which is presented as part of this thesis can be found in Chapter 5. Chapter 6 presents a novel reactive approach for dealing with obstacles as part of the short course routing procedure.

4.1.3 Manoeuvre execution

The short course routing layer delivers a desired direction for the current boat position and wind conditions in real time. If the actual boat direction deviates from the desired direction, the system adjusts the rudder position in order to bring the boat back to the desired course. In parallel, a second control system assures that there is flow in the sails in order to generate propulsion. If a tack or a jibe is required due to significant changes in the desired boat direction, this layer has to ensure a smooth execution of the manoeuvres automatically.

A novel fuzzy logic based approach to manoeuvre execution is described in detail in Chapter 7.

4.1.4 Emergency reflexes

The manoeuvre execution layer passes desired rudder and sail movements to the emergency reflex layer. Normally the rudder and sail settings are passed down to the actuators unchanged. Only in the case of an emergency, when the planned movements pose a threat to the safety of the boat does the emergency reflex layer start acting and overrule the requested actions.

These reflex actions include avoiding capsizing in the case of a wind gust or cautious sailing during periods of strong wind.

4.2 Communication system

This section gives an overview of the communication partners involved and a detailed description of a multi-stage communication architecture and presents some experiments carried out in the Irish Sea.

4.2.1 Communication partners

Three entities are involved in the communication process: sailing boat, visualisation software (Fig. 4.2), and remote controller.

During normal operation the sailing boat transmits sensor values to the visualisation. The visualisation is a software program which runs on a computer on shore and represents the transmitted data like position or information about the boat's strategy clearly. If required it is also possible to send important instructions or data, such as new target coordinates, obstacle information, or a new desired course from the visualisation to the sailing boat.

In case of emergency, especially during test runs, an ergonomically designed remote control device can be used to overrule the autonomous on-board control of the sailing boat. In this case the desired actuator values, including the position of rudder and sails are transmitted to the boat in real time.

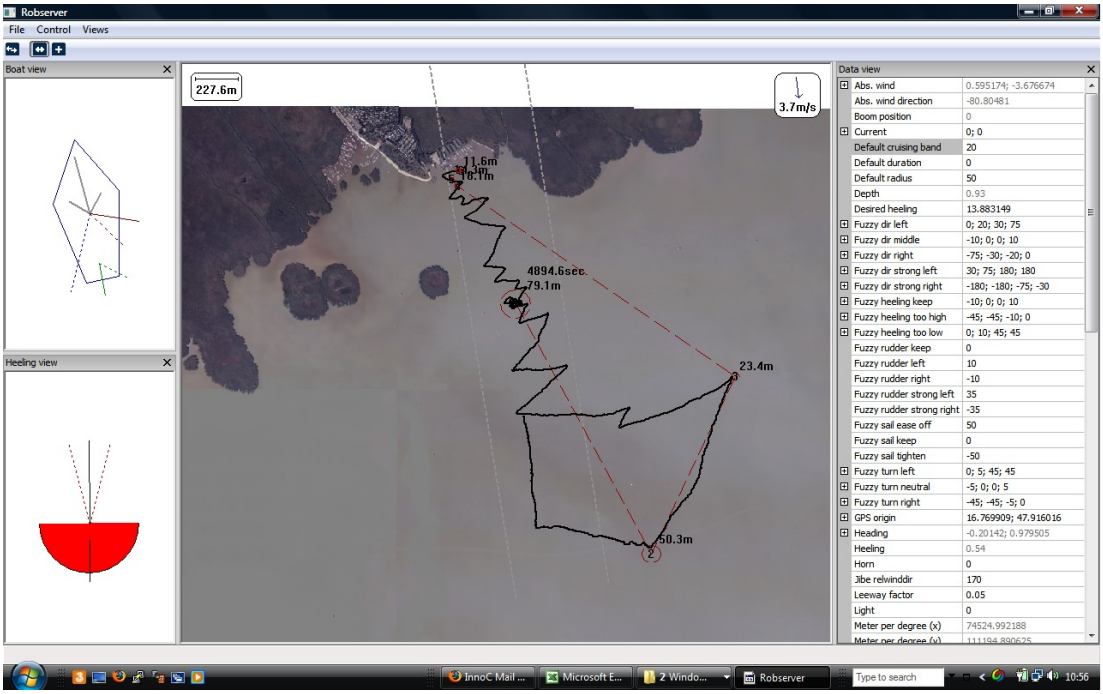


Figure 4.2: Screenshot of visualisation software

4.2.2 Multi-Stage communication architecture

The System consists of three alternative communication channels between boat and shore. Each of them features specific advantages and disadvantages. Therefore, it has to be ensured, that the system switches dynamically between the available communication channels in order to use the most appropriate channel at any given time.

First stage: wireless LAN

A radio mast, equipped with a directional wireless LAN antenna, is mounted on shore. On the mast top of the boat an omnidirectional antenna is mounted (Figure 4.3). The higher the antennas are mounted, the longer the distance that can be covered by this technology. Experiments have shown a reliable 10 Mb data link between boat and shore for up to 3 km. Both antennas are mounted at a level of about 5 m.

Advantages of this technology are:

- No base fee and no connection fee
- High bandwidth which allows even real-time video transmission
- TCP-based communication; software maintenance can be carried out during runtime

Disadvantages are:

- Infrastructure (antenna, router) needs to be set up
- Limited operating distance

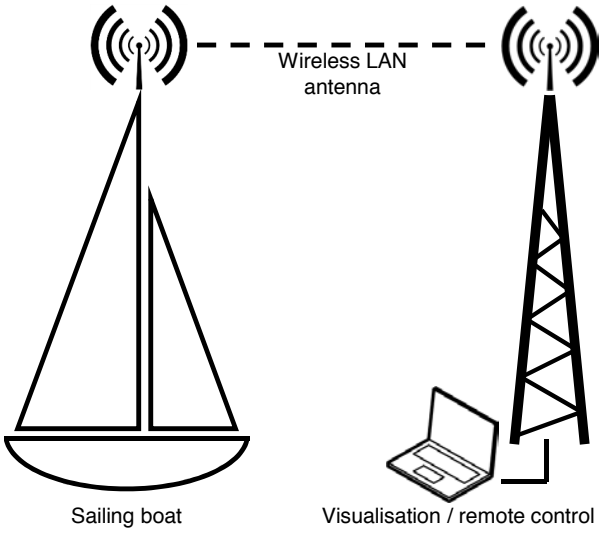


Figure 4.3: First stage - wireless LAN communication

Second stage: 3G data service of mobile phone provider

The boat is equipped as well as a server on shore, with a 3G data modem of a commercial mobile phone provider. 3G or 3rd generation mobile telecommunications, is a generation of standards for mobile phones and mobile telecommunication services fulfilling the International Mobile Telecommunications-2000 (IMT-2000) specifications by the International Telecommunication Union (Smith

& Collins [2002]).This allows internet-based communication between these two stations. Common data modems provide UMTS (Universal Mobile Telecommunications System) and GPRS (General Packet Radio Service) and switch over automatically depending on the availability of these services.

Advantages of this technology are:

- Infrastructure is provided by the mobile phone service provider
- High bandwidth (up to 57.6 kbit/s for GPRS respectively up to 384 kbit/s for UMTS which allows even real-time video transmission)
- TCP-based communication; software maintenance can be carried out during runtime

Disadvantages are:

- Base fee and connection fee, can be extremely high abroad (roaming)
- Operating distance is limited by the network coverage of the service provider[1]

Both communication partners are connected to the internet via a 3G modem. The communication partners are connected via a virtual private network (VPN). (Figure 4.4)

Third stage: satellite communication

The sailing boat is equipped with an Iridium satellite transceiver. The Iridium satellite constellation is a system of 66 active communication satellites with six spares in orbit and on the ground. It enables worldwide voice and data communications. The Iridium network is unique in that it covers the whole earth, including poles, oceans and airways.

Various Iridium based services are available. The most appropriate with communication to an autonomous sailing boat is the Iridium Short Burst Data (SBD) Service. Iridium SBD Service is designed to serve a range of applications that

[1] According to GSM, which is the basis for GPRS communication, technical specification the maximum distance between mobile station and base station is 35 km. (Medved & Tekovic [2004]). However, a viable communication link can be expected up to a distance of about 15 km as shown in Alejos *et al.* [2007].

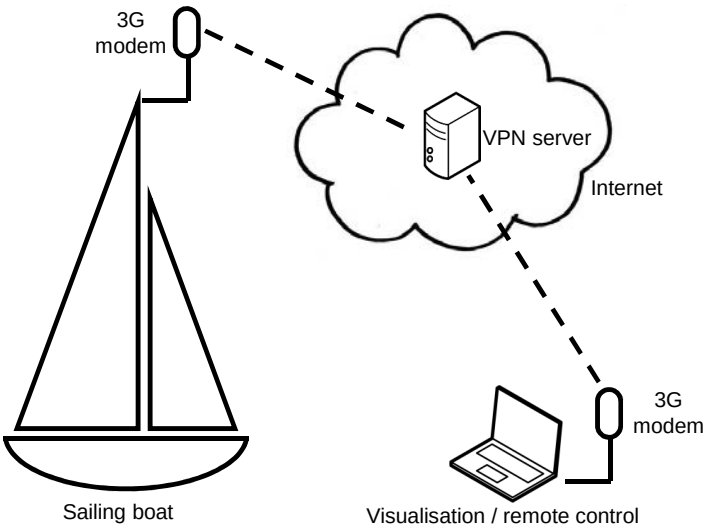


Figure 4.4: Second stage - 3G communication

need to send data messages that on average are typically less than 300 B. The Iridium 9601 SBD transceiver used on the sailing boat is controlled by AT commands over an RS232 serial port. An Iridium Gateway allows receiving data from respectively transmitting data to the sailing boat via e-mail (Iridium [2003]).

Advantages of this technology are:

- Covers the whole surface of the earth
- The Iridium SBD transceiver offers not just message transmission, but delivers rough geographical position information as well. This can be used as a backup system if the GPS receiver on board fails.

Disadvantages are:

- Low data volume: max. about 2 kB per message (Iridium [2003]), which can be sent approximately every 30 s (Sybrandy [2007])
- High transmission latency: between 7 s and 22 s modem processing time (McMahon & Rathburn [2005])
- Base fee and connection fee: USD 21.00 per month plus USD 1.30 per $1,000$ B[1]

[1]Provider AST Connections Ltd. (2009)

The sailing boat is equipped with an Iridium SBD modem, which transmits data via the Iridium satellite network. Data packages are forwarded to a mail server by the Iridium provider. The visualisation software is connected to the Internet via a 3G data link in order to fetch transmitted data from the mail server. (Figure 4.5)

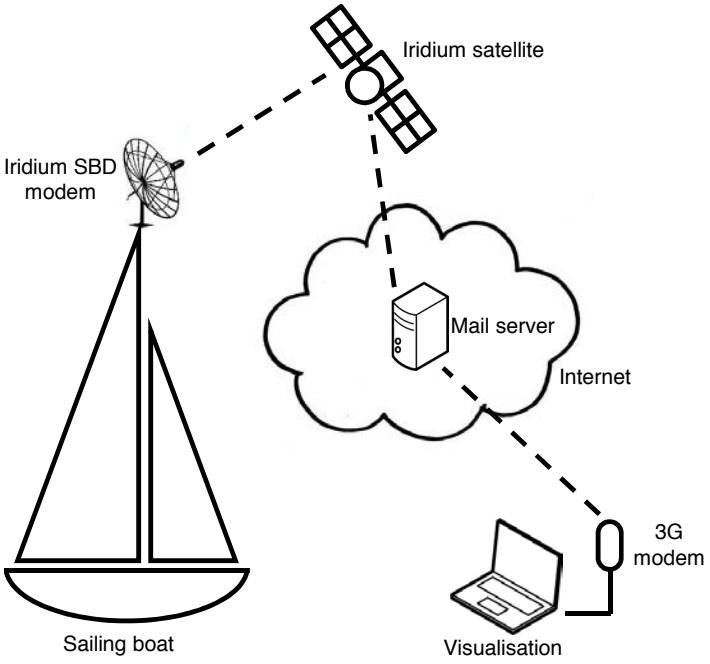


Figure 4.5: Third stage - satellite communication

4.2.3 Selection of the Appropriate Communication Stage

The sailing boat, the visualisation software, and a remote controller are communicating with each other. Basically, three different communication stages (WiFi, 3G, Iridium) are available, but not every communication technology is adequate for every combination of communicating entities. Any of the three stages can be used between sailing boat and visualisation, albeit with a different scope of operation. Remote control requires a line of sight to the boat and real time transmission of the instructions. Therefore, in this case, only Wireless LAN and 3G are suitable. (Fig. 4.6)

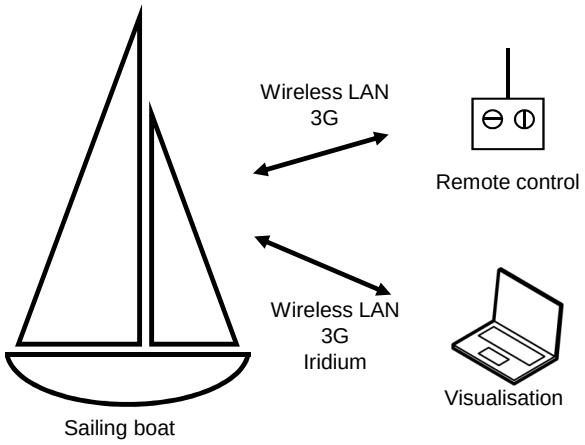


Figure 4.6: Communication partners and available communication stages

The selection of the appropriate communication stage is mainly based on the availability of the data networks. Considering the advantages and disadvantages of each technology the following strategy is implemented.

If a proper wireless LAN link to the boat is available, it will be used for visualisation and as remote controller. If the link quality decreases below a certain threshold, the system automatically switches over to 3G if available. For both communication partners this happens transparently. If neither wireless LAN nor 3G is available, satellite communication via Iridium SBD will be activated. In this case direct manual control of rudder and sail position is not possible and not reasonable. This is because of the limited sight caused by the long distance to the sailing boat and the high latency of satellite communication.

The availability of all communication stages is verified periodically. If indicated, a switch over will be initiated. A hysteresis condition is implemented in order to avoid constantly switching between the communication stages.

When no communication stage is available, the boat keeps on sailing fully autonomously. The sailing functionality of the boat is completely independent from the communication system. A user on shore is not able to monitor the boat's data or to influence its strategy in this situation.

4.2.4 Features of the individual communication stages

Wireless LAN and 3G enables the user to request all available data from the sailing boat. There are two different modes:

- **Pull mode:** a request for a certain value is sent to the sailing boat and immediately answered.
- **Push mode:** the client software can subscribe to a set of values. The sailing boat delivers every update of the subscribed values automatically.

These communication stages also allow transmission of multimedia data, such as images, videos, or sound files. Furthermore it is possible to remotely log into the boat's main computer. This can be used to check log files, update software, or to restart system services during runtime.

The third communication stage, satellite communication, has restrictions concerning bandwidth and latency. Therefore it focuses on transmission of short and concise data packages. By default only the location data (GPS coordinates) of the sailing boat are transmitted regularly. Further values (wind speed, battery voltage, etc.) are transmitted if certain thresholds are reached. As Iridium SBD provides communication in both directions any other value can be requested on demand and high level commands such as new waypoints can be submitted.

4.2.5 Experiments

Experimental setup

The communication architecture described above was extensively tested in September 2007 at the second Microtransat Challenge in Aberystwyth, UK during a 24 h race (see Section 2.2.2). The test runs were carried out with the autonomous sailing boat *ASV Roboat* (see Section 3.2).

The experimental set-up provides all three communication stages presented above. The technical infrastructure was set up as illustrated in Figures 4.3–4.5.

Results and discussion

Figure 4.7 compares the network coverage of all three communication stages in Aberystwyth, where Microtransat 2007 took place. For runs near shore, wireless LAN coverage was sufficient and 3G served as a backup system. Whenever the sailing boat left the area covered by wireless LAN the system switched to 3G automatically without user intervention and vice versa.

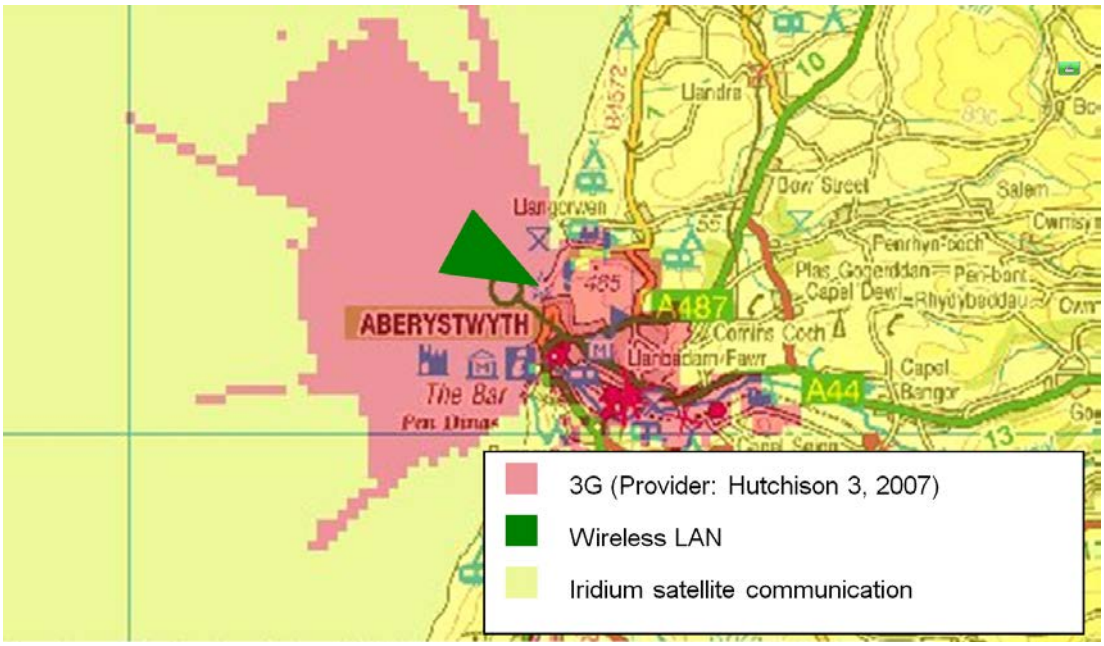


Figure 4.7: Network coverage in Aberystwyth (UK), Microtransat 2007

As there was no race outside 3G coverage, the 3G modem was deactivated manually to test the switchover to satellite communication. As soon as the boat sailed out of the area covered by wireless LAN it switched over directly to satellite communication, because 3G was not available. In this stage the system transmitted the boat's GPS position every 5 min as configured. Remote control of the boat was not possible until the boat sailed back autonomously and received a wireless LAN signal again.

Power consumption is a very important feature for autonomous sailing boats especially on long term missions. Therefore power consumption of the communication equipment of the individual stages is compared in Table 4.1. Since stages

I and II are primarily used for test runs and short missions close to shore, power consumption is not of particular interest. Therefore for long term missions only stage III is incorporated into the energy balance.

Communication Device	Power Consumption
Wireless LAN (Buffalo WHR-HP-54G)	3.4 W (Buffalo [2009])
3G (HUAWEI E220)	$\leq$ 2.5 W (Huaweii [2009])
Satellite (Iridium SBD 9601)	1.75 W during message transfer 0.33 W stand-by (Iridium [2009])

Table 4.1: Comparison of power consumption of communication devices used

4.3 Summary

A four layered hybrid subsumption architecture is presented for an autonomous sailing boat. The system allows autonomous sailing, where routing, navigation and carrying out manoeuvres occur automatically and directly on board the boat. Various sensors observe the highly dynamic environment and provide measured data to the control system, which steers rudder and sails. The four layers are responsible for strategic long term routing, short course routing, manoeuvre execution, and emergency handling. All four layers are executed asynchronously in parallel. They access sensor data directly and generate prerequisites for the succeeding, subordinate layer. This control architecture has been implemented on both sailing robots developed during this research project. Successful test runs and participation with different boats in several international robotic sailing competitions have demonstrated the feasibility and suitability of the presented approach.

Although an autonomous sailing boat can operate without human intervention a data link between boat and shore is necessary. During development a reliable connection for monitoring, debugging, and remote control in case of emergency is essential. Many long-term observation tasks require measurement data to be sent to the operator on shore in real-time. A three-stage communication system for autonomous sailing boats has been designed, implemented and tested successfully.

It combines wireless LAN, 3G and satellite communication ensuring a reliable data link between shore and sailing boat.

Chapter 5

Short course routing

This chapter presents a novel method to calculate a suitable route for a sailing boat in order to reach any specified target destination. The proposed approach does not need a weather forecast at all. As the local wind conditions often change and accurate weather forecasts are not available for very short distances and periods, only local and present wind conditions are taken into account in order to determine an optimum heading for the vessel. The method reacts to changes in the wind conditions in real-time by recalculating the heading periodically.

First, the boat's behaviour is described and the basic principles of the routing method are presented before reporting the flow chart of the algorithm. The particularities of the proposed strategy are illustrated by simulations using a computer model of a sailing boat. Finally, the algorithm is tested on an unmanned autonomous sailing boat equipped with an on-board computer system and the necessary sensor and actuator devices.

5.1 Routing strategy

5.1.1 Local coordinate system

For means of illustration and convenient use of vector operations, local Cartesian coordinates are used to describe the navigated water surface. The simplified assumption implies that the water surface is considered to be flat. This is a good approximation for distances covered by short course routing. The point of

origin of the local system is set somewhere close to the route, e.g. to the starting point or to a reference point nearby on shore. The transformation between the geographic position and the local system is defined as follows:

$$x = R_E \cdot cos\,(LAT) \cdot \frac{\pi}{180\,\mathrm{deg}} \cdot LON \tag{5.1}$$

$$y = R_E \cdot \frac{\pi}{180\,\mathrm{deg}} \cdot LAT \tag{5.2}$$

This means that the x-axis always leads to the east while the y-axis leads to the north. The conventional way of drawing x,y-charts therefore results in the conventional view as seen on northern hemisphere maps where north is towards the top and east is on the right hand side. It is important to notice that the transformation to Cartesian coordinates is done mostly for means of illustration. The navigation strategy can be formulated in a similar way for geographic coordinates using great circle routes, compass headings, and trigonometric functions instead of straight lines, normalized vectors, and vector analysis. According to the following description, the boat virtually moves on a flat water surface disregarding the Earth's curvature.

5.1.2 Sailboat behaviour (polar diagram)

The actual speed a sail boat can reach in a certain direction depends on the wind speed but also on the angle between boat heading and wind direction: while no direct course is possible straight into the wind, the maximum speed is usually obtained with the wind from the rear at about ±120 deg. This dependency can be plotted continuously as a boat-specific polar diagram (Fig. 5.1).

The boat speed is therefore given as a function of the wind speed and the angle between true wind and boat heading:

$$|\vec{v}_b| = f\left(|\vec{w}_{abs}|\,, |\varphi\,(\vec{v}_b) - \varphi\,(\vec{w}_{abs})|\right) \tag{5.3}$$

The polar speed diagram normally shows the norm of the boat velocity vector. Lateral forces caused by the wind lead to a lateral drift, called leeway[1]. The

[1]Leeway is the amount or angle of the drift of a ship to leeward from its heading.

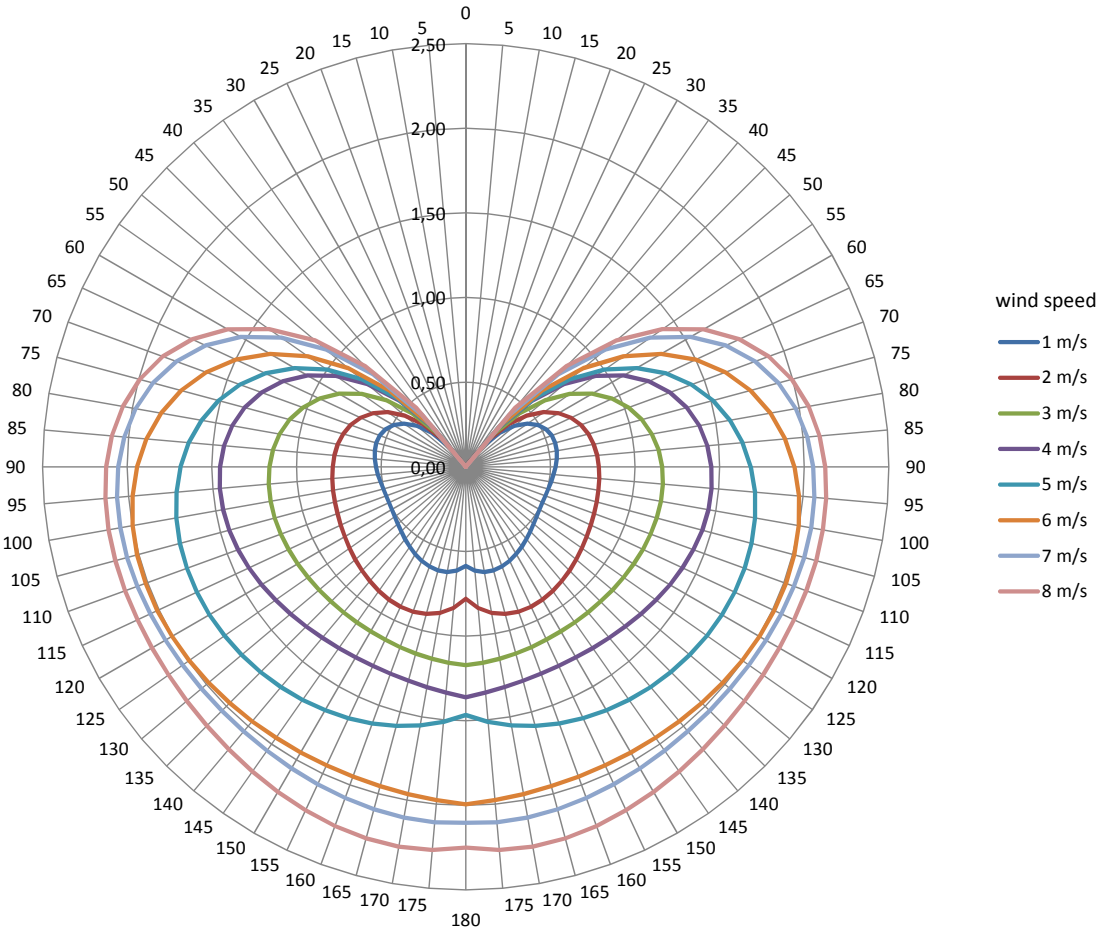


Figure 5.1: Example of a polar diagram produced after a 7 h test run with ASV Roboat on Lake Ontario, Canada in 2010. It shows a velocity prediction in m/s for a particular boat for true wind angles from 0 deg to 180 deg and true wind speeds up to 8 m/s. Note that the speed drops to zero as the boat heads closer to the wind.

heading of the boat is therefore always slightly closer towards the wind than the actual direction of movement. As leeway is a very important factor in sailboat route planning, the leeway drift behaviour of the sailboat needs to be considered. However, the diagram in Fig. 5.1 does not include information about the difference in directions of boat heading or actual movement. Additional information about leeway drift as a boat dependent function of wind speed and direction is required.

5.1.3 Quantification of target-approach

In order to get from a current position of the boat B to a target point T, both the direction of the target and the wind must be considered. The aim of the routing algorithm is therefore to decrease the distance to the target as fast as possible[1]. The efficiency of a particular boat heading when approaching the target can be directly quantified by projecting the boat speed vector on the target direction:

$$v_t = \vec{v_b} \cdot \vec{t_0} \tag{5.4}$$

Eq. 5.4 is illustrated in Figs. 5.2 and 5.3under various wind conditions. The boat speed vector $\vec{v_b}$ can be considered to be a function of the boat heading $\vec{v}_{b,0}$ and the wind vector according to Eq. 5.3.

5.1.4 Beating hysteresis and beating parameter

In practice, the sailor beats about if the target is within the angle where no direct navigation is possible. In terms of our analytical approach, this means that the boat follows a local optimum $\vec{v_b}$ (close to the recent heading) for a certain time until the global optimum $\vec{v_b'}$ is significantly better than $\vec{v_b}$. At this point, the boat turns towards the global optimum $\vec{v_b'}$, which will be followed until an alternative heading proves to be significantly better leading to the next turn and so on. Beating is illustrated in 5.4, the hysteresis factor n is defined by:

$$v_t' > n \cdot v_t \rightarrow \text{turn for } \vec{v_b'}; n > 1 \tag{5.5}$$

[1]The speed of a sailing boat relative to the waypoint T it wants to reach is also referred to as velocity made good (VMG)

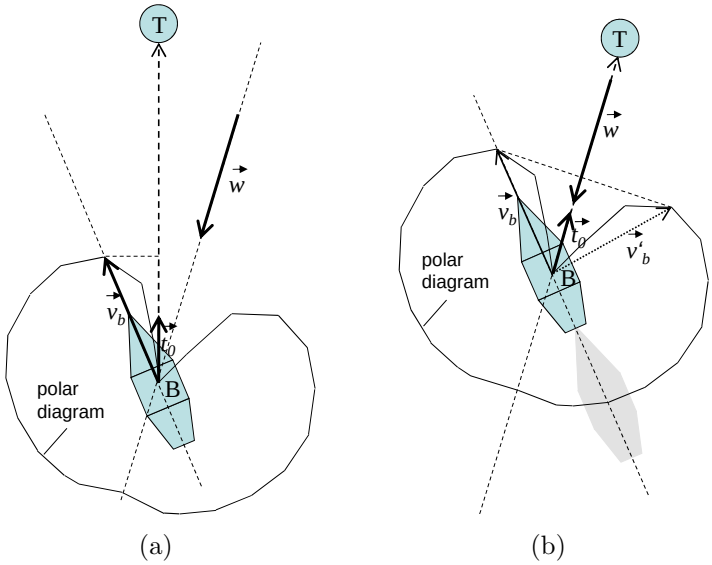


Figure 5.2: Determination of optimum heading on upwind course: Fig. (a) shows a situation where the target is located in the direction the wind is coming from. The optimal route is a compromise between aiming towards the target and sailing fast. The purpose of the routing algorithm is to identify the boat heading for which the velocity made good v_t, which represents the negative time-derivative of the distance between boat and target, is maximised. However, the optimal boat heading indicated by the direction of the speed vector changes as the boat moves on its trajectory. Fig. (b) illustrates the situation slightly later on the course. The grey shadow of the boat indicates the boat's position at the earlier point in time which is described in (a). For the situation in Fig. (b) there are two headings of equal maximum velocity made good to follow, one on the right and one on the left hand side of the wind direction. This happens when the target direction aligns with the wind direction. In order to get a unique proposal for the heading to follow, a hysteresis condition is applied.

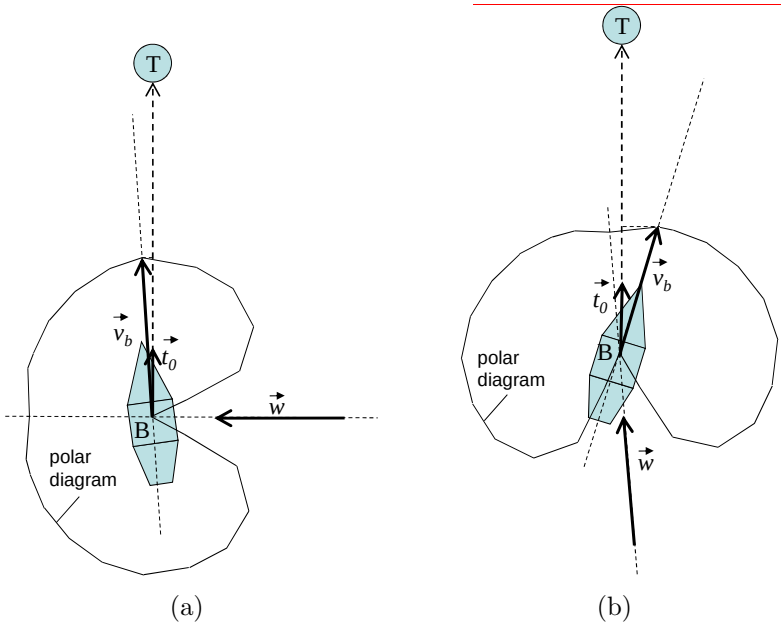


Figure 5.3: Determination of optimum heading on beam reach and downwind course: Figs. (a) and (b) illustrate, that the same approach as for upwind courses (see Fig. 5.2) works if the target is located in any direction relative to the wind direction. Those two examples promise unique identifiers for the optimum boat heading until the target is reached and the steady correction of the boat heading is smooth along the trajectory.

In order to obtain a reasonable behaviour of the algorithm, n must be larger than one. It can be shown that a constant hysteresis factor leads to a sector-shaped beating area with an increasing frequency of turns in the vicinity of the target. In order to obtain a more or less rectangular beating area of defined width, the hysteresis parameter n is expressed as a function of the distance to the target as follows:

$$n = 1 + \frac{p_c}{|\vec{t}|} \tag{5.6}$$

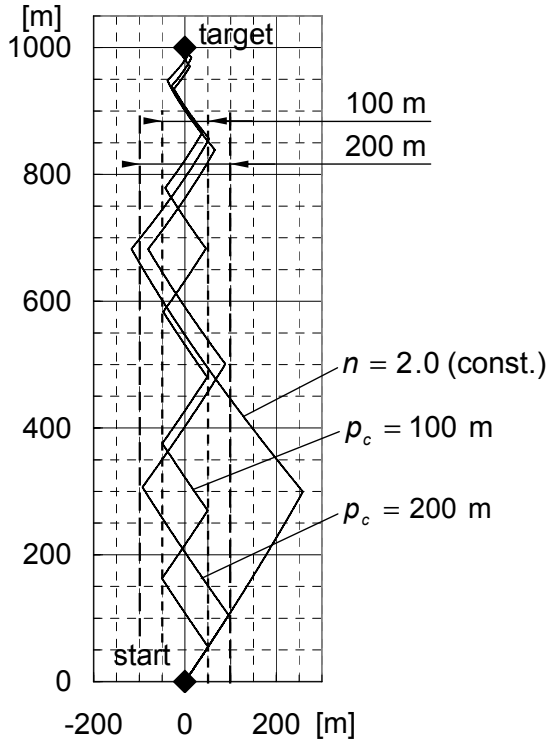


Figure 5.4: Effect of hysteresis factor on beating area

The constant beating parameter p_c in Eq. 5.6 has the dimension of a length and is proportional to the width of the rectangular beating band in adequate distance from the target. For the polar diagram chosen for generating Fig. 5.4, the width of the beating band coincides with the value of the beating parameter (proportionality factor of one). Of course, the orientation of this rectangular beating area can change as the wind direction may change in time. It is important

to note that the beating hysteresis can be globally applied and works without adaptation also in the case where the target is in the direction towards which the wind is blowing (Fig. 5.2). The time-losses of manoeuvres are not considered in the conditions for tacking (Eq. 5.5 and Eq. 5.6).

Simulation shows that all three courses in Fig. 5.4 need the same time to reach the target if manoeuvre costs are not considered. This is plausible because the mean boat heading against the wind is constant regardless of the beating band width. Optimisation with respect to manoeuvre losses would always lead to a course with only one tack. However, a route with only one tack requires a lot of lateral space and is less flexible regarding spontaneous changes in wind direction. Theoretically, if manoeuvre losses are disregarded, the overall route performance does not depend on the beating band width and the distance travelled is identical. The beating parameter should therefore be chosen as a compromise between the available, obstacle free area and performing a reasonable number of tacks considering possible changes in wind direction.

5.1.5 Leeway drift consideration

If the boat is steered in the direction proposed by the optimisation of velocity made good (optimum boat heading), the boat will actually move into a slightly different direction due to leeway drift. The goal of the optimisation procedure, however, is to make the boat move a certain optimum direction rather than to specify the boat heading. To account for leeway drift, the lateral speed component is estimated as a function of the wind vector:

$$\vec{v}_d = f_d \cdot \vec{n}_{b,0} \cdot (\vec{n}_{b,0} \cdot \vec{w}_{abs}) \tag{5.7}$$

The leeway velocity $\vec{v}_d$ is added to the calculated boat speed vector from the boat polar diagram. The dimensionless leeway factor f_d depends on the boat and can be experimentally determined. The boat heading has to be adjusted to $\vec{v}_b - \vec{v}_d$ in order to achieve the boat movement in the optimum direction $\vec{v}$.

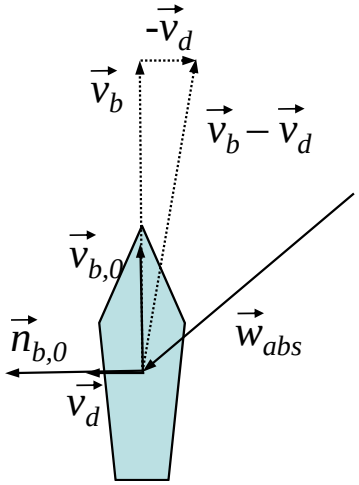


Figure 5.5: Leeway model

5.1.6 Summary of algorithm and implementation

The data needed to decide the boat heading in order to efficiently reach the target are:

- target position T
- current boat position B
- current boat heading $\varphi\,(\vec{v}_b)$
- true wind direction and speed (i.e. wind vector $\vec{w}_{abs}$)

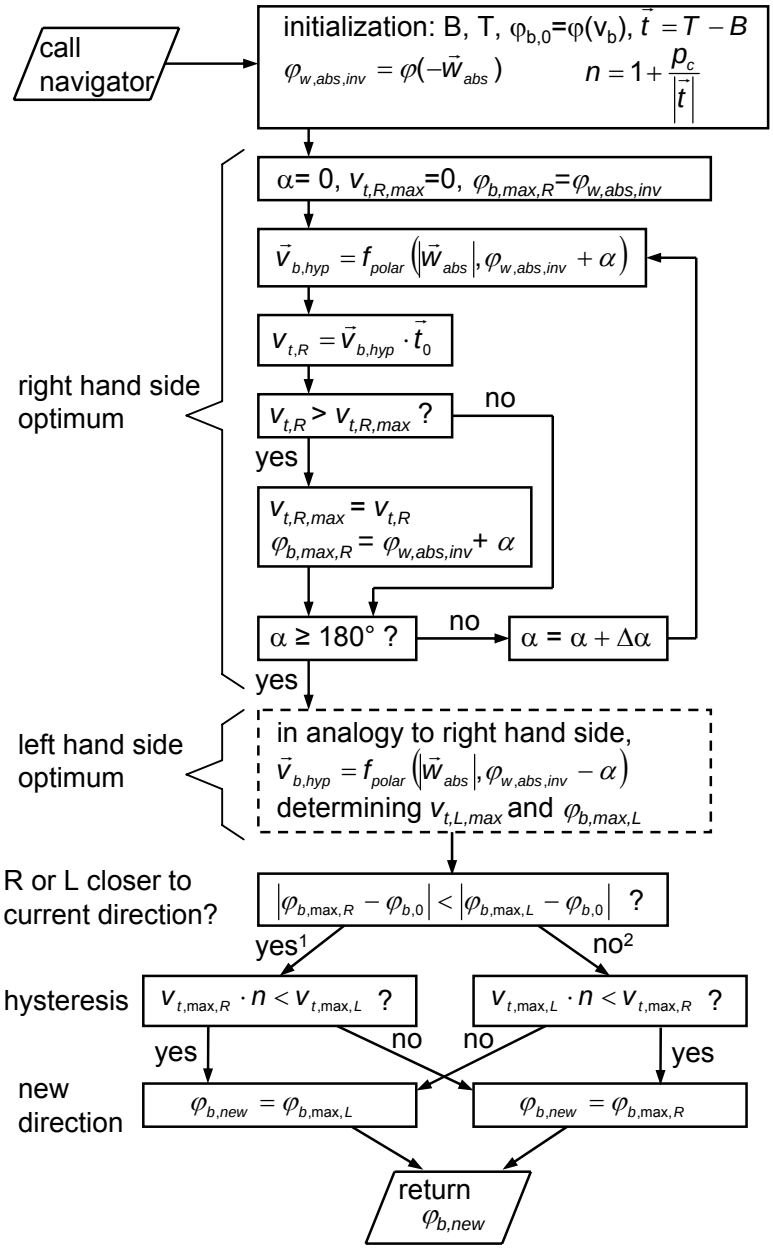
The current boat heading is needed in order to decide whether a proposed direction requires a tack or not. The wind speed is only needed if the polar diagram shows a significantly non-linear dependency on the wind speed. Otherwise, only the direction of the true wind is actually required and a normalised polar diagram is used. At this point we assume that the necessary data are available. The practical determination of the true wind from different sensor values will be treated in the experimental section below.

The routing algorithm shown generally assumes that

- the true wind is the same all over the area between boat and target and
- the true wind will remain for the whole leg as it is at the moment (or as it has been over a recent time interval for average wind determination respectively)

The first assumption is reasonable for small to medium range environments like lakes or coastal regions and the second assumption reflects the knowledge of a sailor paying no heed to the weather forecast.

The overall structure of the routing algorithm is illustrated in Fig. 5.6. The aim is the calculation of a suitable boat heading in the form of an angle (which can subsequently be transformed to a unit vector) for a given parameter set. The routing algorithm is called again and again in reasonable time steps in order to update the proposed heading as the surrounding parameters change. This applies especially to the target direction as a result of boat movement.



1) right hand side optimum closer to current boat direction
2) left hand side optimum closer to current boat direction

Figure 5.6: Structure of the short course routing algorithm

5.2 Experiments

5.2.1 Experimental setup

Experiments on the routing algorithm were carried out on a computer simulation first, and afterwards on the 1.38 m yacht model *Roboat I* (see Section 3.1 for a detailed description). The *Roboat I* is usually used as a remote controlled sailing boat. For our purposes, the boat is additionally equipped with various sensors to measure the environmental conditions. A computer program called *Abstractor* running on the boat gathers sensor data and transforms them into semantically useful information for the routing software.

The target position T is statically defined for a particular leg and does not need to be measured. The current boat position B is measured by a GPS receiver. The Abstractor transforms the GPS coordinates into the metric Cartesian coordinate system. A tilt compensated compass delivers the boat heading $\varphi(\vec{v}_b)$. Because of the inaccuracy of the sensor data some damping is applied. The true wind $\vec{w}_{abs}$ has to be calculated out of the apparent wind $\vec{w}_{rel}$ and the boat velocity $\vec{v}_b$ (Eq. 5.8 and Fig. 5.7), where $\vec{v}_b$ is a combination of $\varphi(\vec{v}_b)$ and actual boat speed $|\vec{v}_b|$ measured by the GPS receiver.

$$\vec{w}_{abs} = \vec{w}_{rel} + \vec{v}_b \tag{5.8}$$

5.2.2 Results and discussion

Since the shapes of the curves for varying wind speeds are largely concentric (Fig. 5.1), a normalised shape-function multiplied with the true wind speed is used to describe the polar diagram of the boat for the present work. The coefficients $a_0 ... a_5$ in Eq. 5.9 for α in deg are listed in Table 5.1 and the resulting graph is shown in Fig. 5.8.

$$|\vec{v}_b| = max\left(0, |\vec{w}_{abs}| \cdot \sum_{k=0}^{5} \left(a_k \cdot a^k\right)\right) \tag{5.9}$$

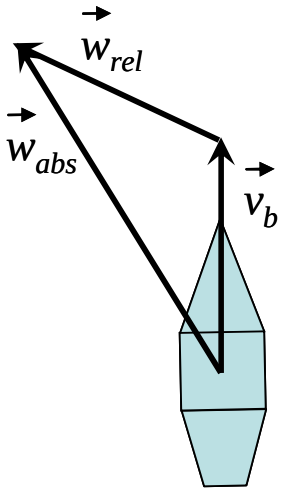


Figure 5.7: True and apparent wind

Coefficient	Value	Unit
a_0	-1.3956	-
a_1	$1.0786 \cdot 10^{-1}$	deg^{-1}
a_2	$-2.3250 \cdot 10^{-3}$	deg^{-2}
a_3	$2.4255 \cdot 10^{-5}$	deg^{-3}
a_4	$-1.1939 \cdot 10^{-7}$	deg^{-4}
a_5	$2.2054 \cdot 10^{-10}$	deg^{-5}

Table 5.1: Coefficients for the normalised polar diagram (according to Eq. 5.9)

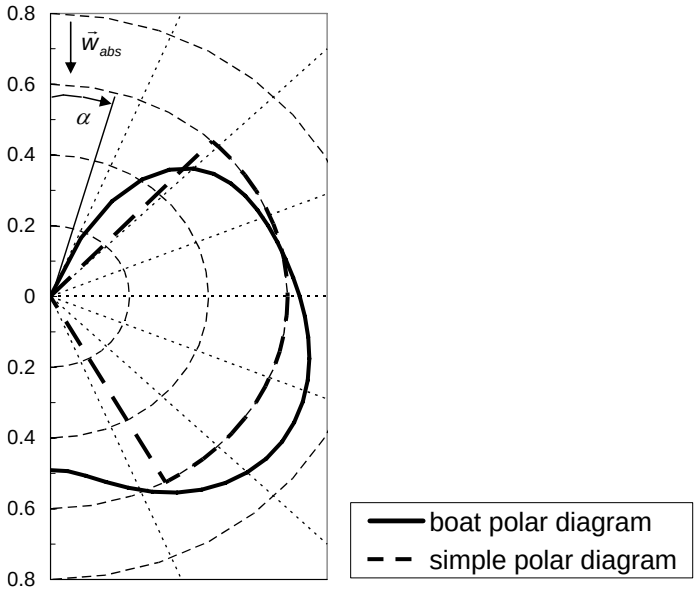


Figure 5.8: Normalised polar diagram used within the present work (according to Eq. 5.9 and Table 5.1 for unit wind speed)

The polar diagram according to Fig. 5.8 is used to describe the boat's behaviour in the following computer simulations. The routing algorithm uses either this polar diagram or a binary simplification returning a constant average speed for courses between 43 deg and 151 deg (broken line in Fig. 5.8). The boundaries of the simple polar diagram have been chosen based on time-optimum simulation results. Several boundary angles have been simulated. The values taken led to the shortest run times on the evaluated courses. It is important to notice that the consideration of leeway drift in the routing algorithm leads to boat headings closer to the wind than the proposed optimal course. The closest proposed direction of boat movement against the wind is predefined by the shape of the polar diagram used in the routing algorithm. To avoid getting stall against the wind, these closest courses must leave room within the navigable range for heading adjustment due to leeway drift compensation. This requirement has a direct impact on the shape of an efficient polar diagram, which can, as it will be shown below, differ from the actual boat polar diagram.

For a simple course connecting two points, the routes proposed by the routing algorithm are shown in Fig. 5.9 for different constant wind conditions. The

beating parameter is set to 60 m for all runs. The mathematical model of the boat behaves strictly according to the boat polar diagram without consideration of leeway. In each case, the use of either the boat polar diagram or the simple polar diagram for the routing algorithm is compared.

If the target is located upwind (Fig. 5.9(a)), the proposed routes are almost the same for both cases. The boat must tack several times and the approximately constant width of the beating band can be observed in the illustration. In those situations where the wind blows laterally within the navigable range (Figs. 5.9(b) to 5.9(d)), a straight line route is supported by the algorithm featuring the simple polar diagram while the consideration of the boat polar diagram leads to deviations from the straight line. The reason for this behaviour is that higher target efficiencies can be reached according to the boat polar diagram if the boat heading deviates from the straight line. However, this optimisation is only true for that specific moment in time and does not take into consideration possible disadvantages later on the course. In Fig. 5.9(e), where the wind blows straight towards the target, three possible routes are compared. The algorithm suggests routes where the boat jibes several times for both polar diagrams (boat and simple). The third possibility considered in Fig. 5.9(e) is the straight line.

In order to quantitatively compare the different routes in Fig. 5.9, the reciprocal target approaching velocity (reciprocal value of velocity made good) is plotted versus the distance to the target in Fig. 5.10. In these diagrams, the area below the graphs corresponds to the time needed to reach the target (time effort). The time efforts of the different routes are summarised in Table 5.2. The proposed routes do not significantly differ in those situations where the target is located upwind (Fig. 5.10(a)). If the wind is blowing laterally, it turns out that the straight line route proposed by application of the simple polar diagram in the routing algorithm is advantageous at least for the wind directions investigated. Figs. 5.10(b) to 5.10(d) show the time effort where the direct route is most efficient. This seems to be paradoxical at first sight because the use of the actual boat polar diagram in the routing algorithm always follows the direction of highest velocity made good. However, the boat takes a longer route and manoeuvres into areas where the target cannot be approached efficiently any more.

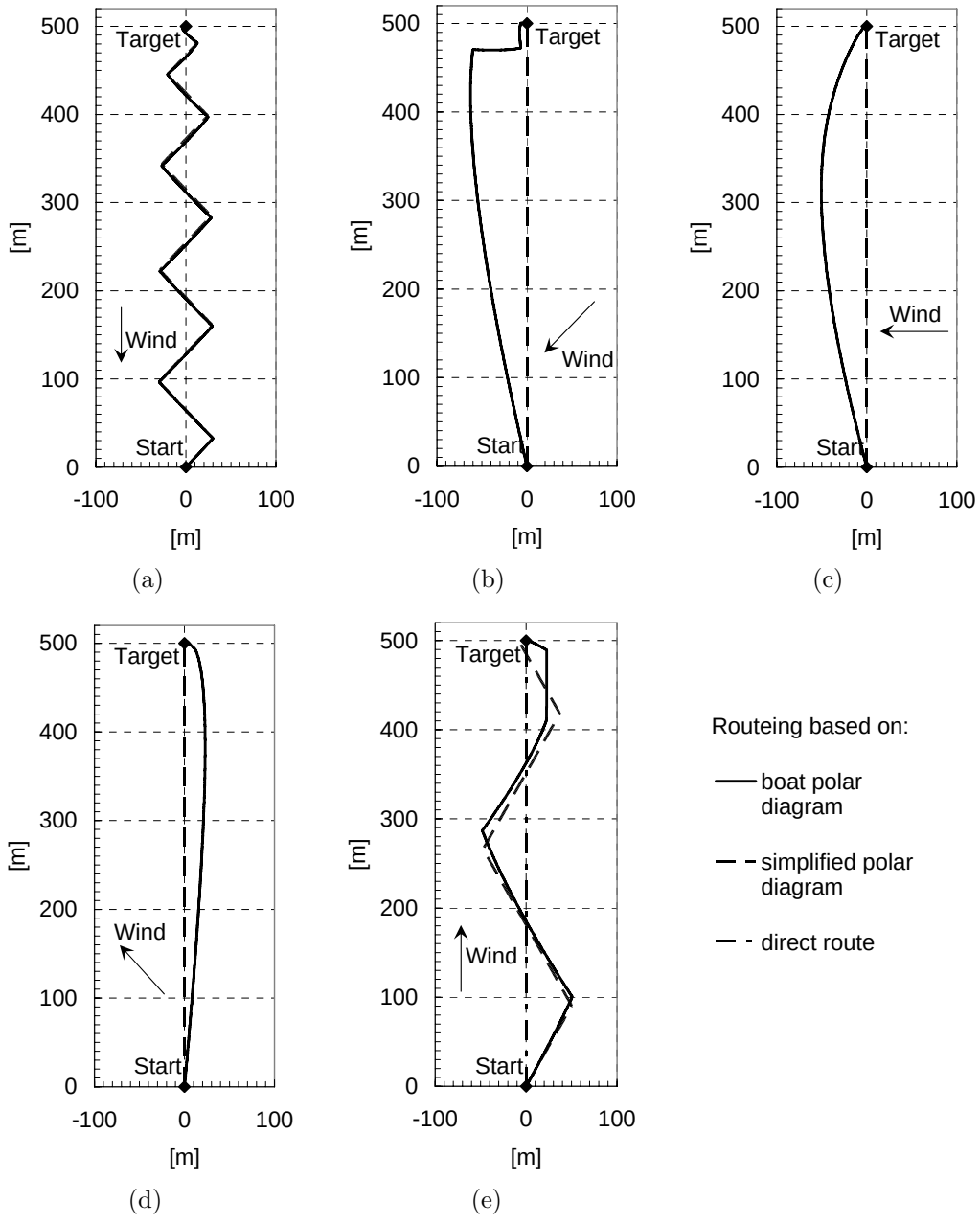


Figure 5.9: Routing simulation results for different constant wind directions

Fig. 5.11 illustrates how the application of the boat polar diagram leads to such a non-optimal route. The diamonds in Fig. 5.11 show the position of the boat in certain points in time on the straight line route (B_{tn}) and on a sub-optimal route based on the boat polar diagram (B'_{tn}). From the very beginning, the proposed non-direct route allows a higher velocity made good. After time step 6, the direct route is already closer to the target than the route with local optimisation of velocity made good. In addition, the boat position of the non direct route at time step 6 is such that the target is in an unfavourable direction with respect to the wind.

In the case where the target lies directly in the wind direction (Fig. 5.10(e)) the proposed routes are both better than the straight line route. The reason is the characteristic shape of the boat polar diagram, where the maximum speed is reached at angles between 120 deg and 140 deg from the direction the wind is coming from (broad reach course).

Time Effort in s	**Wind Direction in *deg*** (0 *deg* in Positive x-Direction)				
	270	**225**	**180**	**135**	**90**
Boat Polar Diagram	2850	2175	1680	1510	1930
Simple Polar Diagram	2835	2010	1630	1460	1860
Straight Line Route	*Not possible*	2010	1630	1460	2095

Table 5.2: Time effort for the routes discussed in Fig. 5.9 and Fig. 5.10 (polar diagram according to Table 5.1 and unit wind speed)

Summarising, the simulation shows that the routing algorithm does not require the knowledge of the detailed boat speed polar diagram in order to indicate suitable routes. Moreover, the routes proposed by applying the simplified polar diagram from Fig. 5.8 are even more time-effective than those proposed on the basis of the boat polar diagram. However, with a non-greedy deliberative strategy even better results could be obtained using the actual boat polar diagram. The time-effort of a certain route between two points can be illustrated according to Fig. 5.10.

The simulation above assumes that the boat behaves strictly according to the boat polar diagram. In order to prove the practical applicability of the proposed

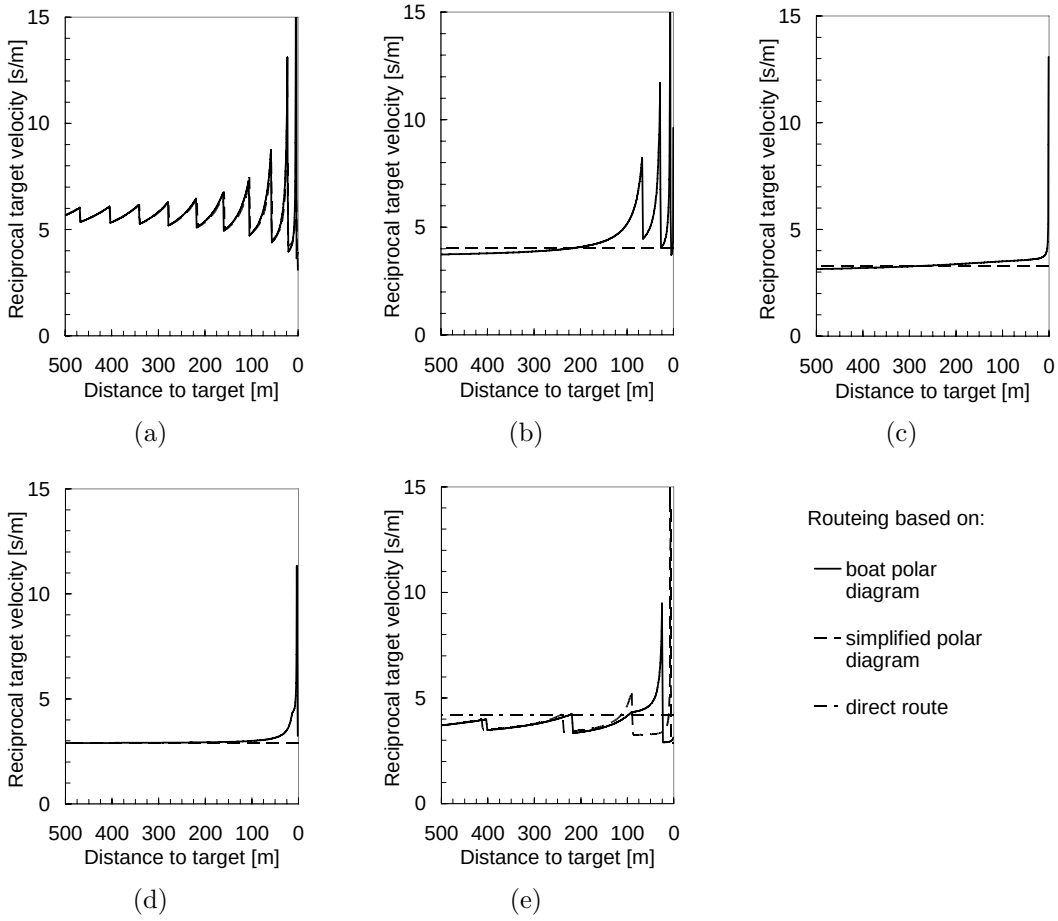


Figure 5.10: Efficiency comparison for the different routes shown in Fig. 5.9

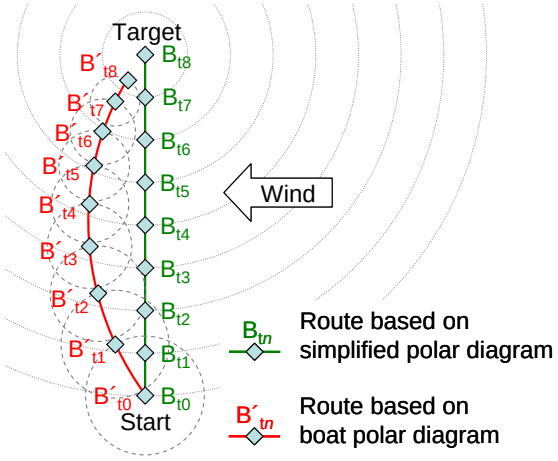


Figure 5.11: Illustration of efficiency comparison on beam reach course (situation as in Fig. 5.9(c) and Fig. 5.10(c))

routing algorithm, test runs have been carried out using the demonstration sail boat *Roboat I*. It is important to notice that the exact polar speed diagram of the demonstration boat is not known and secondary effects like leeway may occur. The presented data refers to the final test run prior to the Microtransat competition 2006 in France, where the wind conditions were within the operational range of the demonstration boat. The wind data during the 20-minute course are plotted in Fig. 5.12.

Fig. 5.13 shows the GPS log data from the test run, where the task was to sail from Buoy 1 to Buoy 2 and back to Buoy 1. The boat polar diagram according to Table 5.1 has been applied for the routing algorithm and the beating parameter has been set to 60 m. It can be observed how the boat enters a beating band before reaching Buoy 2. The experiment is compared to simulation results using the varying measured wind data as an input. The dotted line is the result of a simulation where the boat model behaves strictly according to the boat polar diagram. In reality, the boat behaviour is characterised by lateral displacement according to the wind direction (leeway drift, Fig. 5.5). The dimensionless leeway factor f_d in Eq. 5.7 can be determined by graphical comparison between simulation and experimental data. The broken line in Fig. 5.13 shows the close agreement between the actual data and the simulation for a leeway factor of 0.1. The behaviour of the demonstration sail boat can therefore be accurately

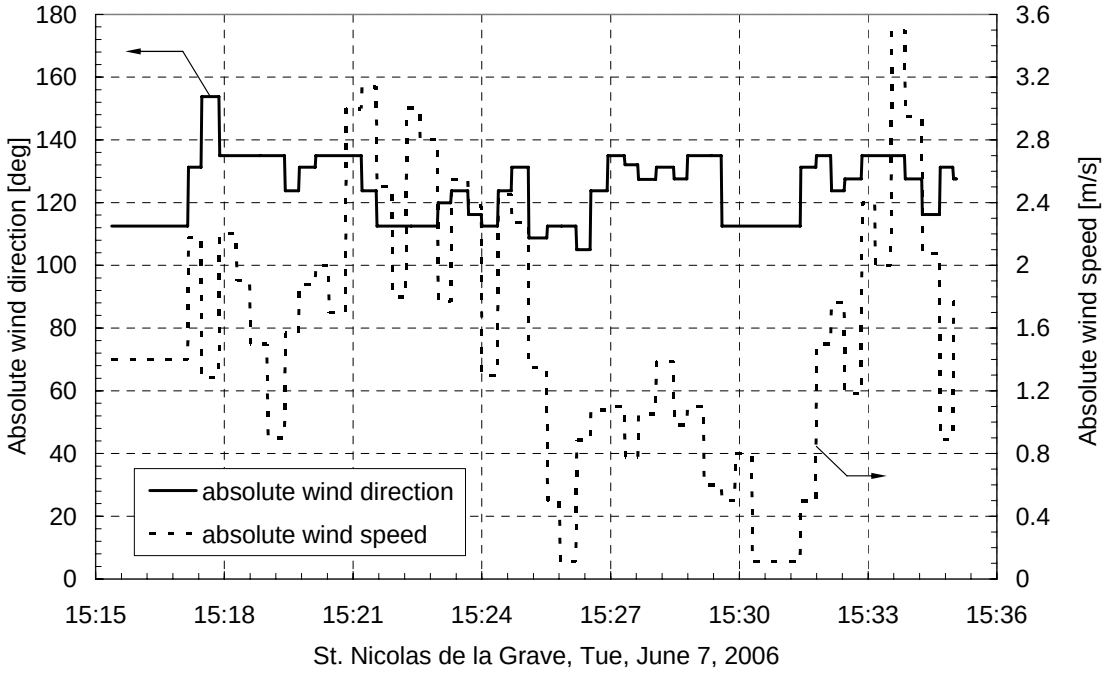


Figure 5.12: Wind log data from the test run

described well by the assumed polar diagram (Eq. 5.9, Table 5.1, re-scaled by multiplication with a factor of 1.21) in combination with the leeway correction (Eq. 5.7, $f_d = 0.1$).

Finally, the application of the simple polar diagram in the routing algorithm is tested and compared to the version using the boat polar diagram. The boat model behaves according to the boat polar diagram featuring the leeway correction. Fig. 5.14 shows that the algorithm featuring the simple polar diagram would have lead to a shorter route. The total time effort for the course is decreased by about 9 % (Table 5.3).

An additional simulation run has been carried out, where leeway is compensated for within the routing algorithm according to Eq. 5.7 and Fig. 5.5. This means, the boat always steers in direction $\vec{v}_b - \vec{v}_d$. Due to leeway drift, the actual movement of the boat changes to $\vec{v}_b$ which is the desired direction of movement originally proposed by the routing algorithm. For the actual wind conditions of the experiment the leeway compensated simulation delivers the third route shown in Fig. 5.14 (dotted line). In this case the runtime further decreases. The devia-

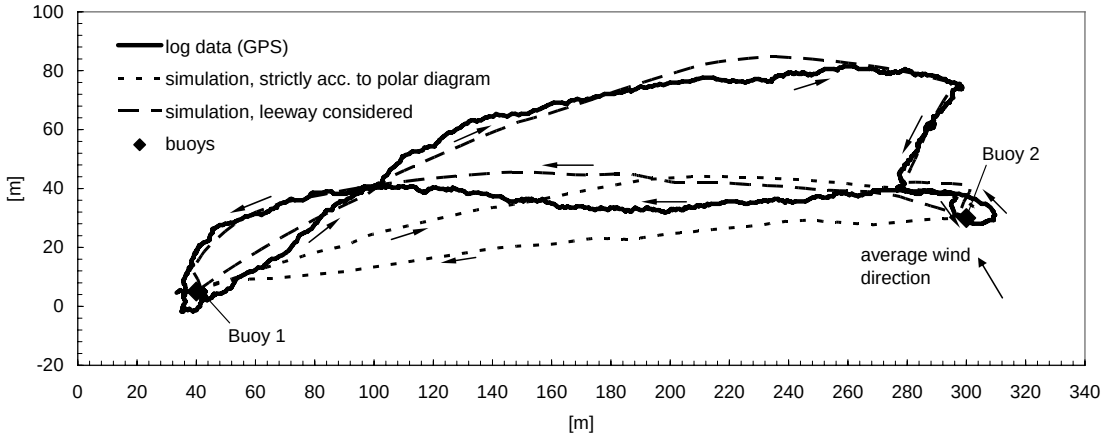


Figure 5.13: Actual run and comparison to simulation results based on real wind data

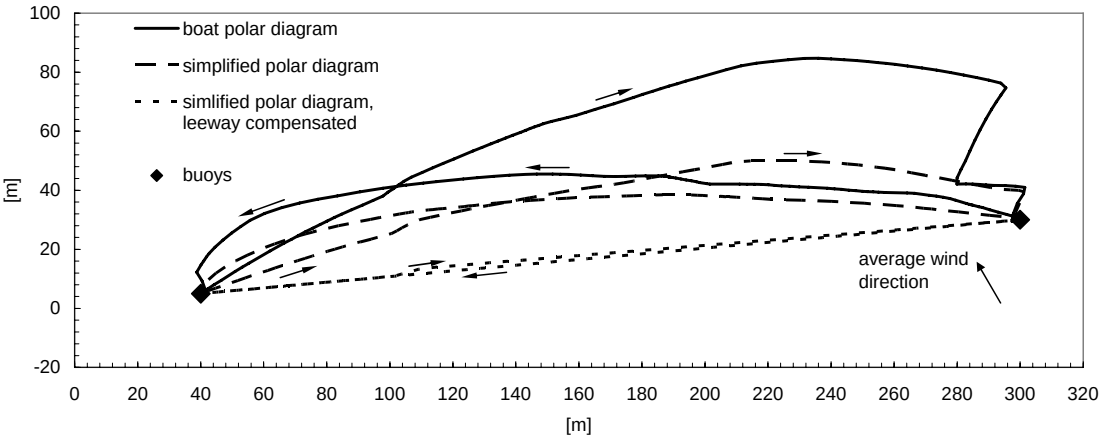


Figure 5.14: Comparison of algorithms featuring either boat polar diagram or simple polar diagram

tion from the straight line route about two minutes after the start is due to the temporary change in wind direction at this point (Fig. 5.12).

Route	Time effort
Boat polar diagram	1135 s
Simple polar diagram	1035 s
Simple polar diagram; leeway compensated	970 s

Table 5.3: Runtime comparison for route in Fig. 5.14

Simulations and test runs based on the actual boat polar diagram have shown a behaviour which is typical for many greedy algorithms where local optimisation for the current point in time cause a penalty later and therefore result in globally sub-optimal behaviour. However, it can be stated that the algorithm finds suitable routes for real wind data also and that, again, a better route is found if the routing algorithm uses the simplified polar diagram instead of the actual boat polar diagram.

5.3 Summary

A compact method to calculate a suitable route for a sailing boat in order to reach any specified target is presented. The calculation is based on the optimisation of the time-derivative of the distance between boat and target and features a hysteresis condition, which is of particular importance for beating to windward. The algorithm provides an answer to the question when to tack on upwind courses. Further, it immediately adapts to varying wind conditions. The resulting routes for different conditions have been analysed on the basis of a simulation featuring a mathematical boat model. Experiments have been carried out using an unmanned and autonomously controlled sail boat. The navigated route closely agrees with the simulation results.

So far most robotic sailing boats simply navigate in a straight line towards the target if possible. If the straight line is not navigable the boat sails a navigable heading pointing as closely towards the target as possible until it is back on a position where it can reach the target directly.

This simple approach works fine when no obstacles need to be taken into account. In this case it shows similar performance to the presented algorithm. The proposed method demonstrates its strengths as basis for the reactive obstacles avoidance method proposed in the succeeding Chapter 6.

Chapter 6

A reactive approach to obstacle avoidance

An important problem to be solved for long-term unmanned and autonomous missions on the sea is reliable obstacle detection and avoidance. Static obstacles such as landmasses can be predefined on the sea chart as a basis for the routing system. A combination of multiple techniques, such as thermal imaging, radar, camera, and automatic identification system (AIS)[1] can be used to detect moving obstacles. Research in this field has been carried out for autonomous underwater vehicles Showalter [2004] and motorised autonomous surface vehicles (Benjamin *et al.* [2006]; Larson *et al.* [2007]; Smierzchalski [2005]; Statheros *et al.* [2008]). The obstacle avoidance task is different for sailing vessels, as they can not navigate directly in any direction, and are dependent on wind conditions. Therefore a more sophisticated approach to autonomous obstacle avoidance is shown.

6.1 Basic idea

To deal with obstacles we extend our *quantified target approach* (depicted in Figs. 5.2 and 5.3) by an *obstacle quality*. There is an overall maximum distance which we call the *safe horizon* r_{max}. Segments beyond it can be ignored. Ob-

[1] Automatic identification system (AIS) is a navigation system for locating, identifying and tracking marine vessels. Maritime laws require AIS on voyaging ships with gross tonnage of 300 or more.

stacles within the safe horizon put a penalty on the directions leading to them. This penalty increases the closer an obstacle. Very close obstacles whose distance threaten to fall below a certain value r_{min} mark corresponding directions as unnavigable.

We distinguish two cases:

- **Unnavigable** courses which cannot be navigated because of wind direction or because an obstacle is too near (see Section 6.5). These courses are not considered in the calculation of an optimal course.

- **Navigable** courses. Considering near obstacles (distance $d_b < r_{max}$ safe horizon) the courses are dynamically modified. Directions leading to an obstacle suffer a penalty q_b as a weight to v_t discussed in Eq. 5.4. q_b is calculated by formula

$$q_b = \min\left(1, \max\left(0, \frac{d_b - r_{min}}{r_{max} - r_{min}}\right)\right) \tag{6.1}$$

  in effect a linear scaling between 0 and 1 within the range $r_{min} \leq d_b \leq r_{max}$ and 0 or 1 outside of it respectively; see Fig. 6.1

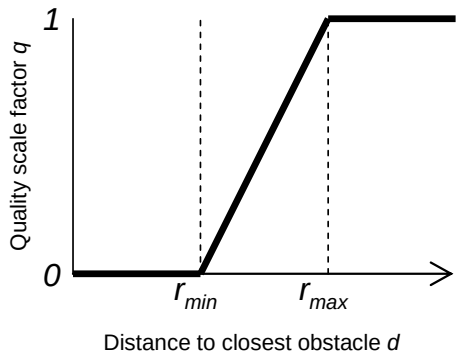


Figure 6.1: Linear scaling on polar diagram according to distance to obstacles

Distinguishing these two cases and using the qualifying weight from the latter we establish a *qualified* v_t^* (in comparison and extension to v_t from Eq. 5.4):

$$v_t^* = \begin{cases} -\infty, & \text{if } |\vec{v_b}| = 0 \\ -\infty, & \text{if } r_{min}\text{-violation (see Section 6.5)} \\ q_b \vec{v_b} \cdot \vec{t_0}, & \text{else} \end{cases} \tag{6.2}$$

A high penalty means a low q_b (close to 0), a low penalty means a value close to 1. This qualified v_t^* allows the course optimization algorithm to compare values of different courses where obstacles are included in the metric as well.

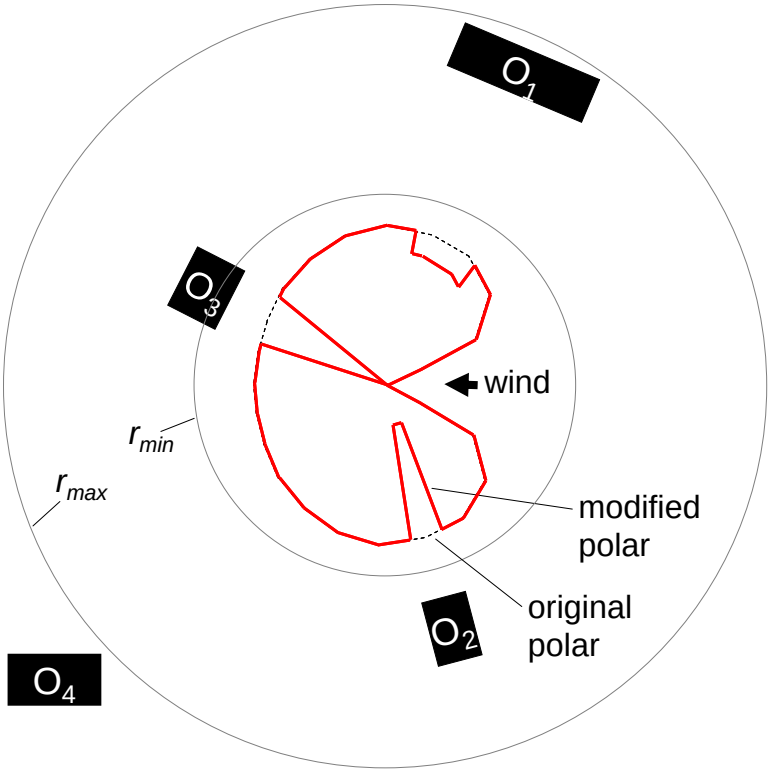


Figure 6.2: Influence of obstacles O_1-O_4on polar diagram

In Fig. 6.2 we show the dynamic modification of the polar diagram. Only obstacles within the safe horizon r_{max} are considered, which omits obstacle O_4. Obstacle O_1 puts only a small penalty to the direction vectors leading to it. The closer obstacle O_2 causes a much higher penalty on its directions. This results in a

dent in the polar diagram. Obstacle O_3 violates r_{min} and therefore the directions towards it get marked unnavigable.

6.2 Obstacle data processing

In collision avoidance a robotic sailing boat must deal with its nearest obstacles. We abstract the boat to a point of view and the obstacles to polygons consisting of line segments. Rays radiating from the point of view, for instance at discrete angles from 0 deg to 359 deg, divide the surrounding area into sectors of equal size. For each sector we determine the minimum distance to the nearest obstacle.

In our context, the result of this calculation is used for close range course planning. For 360 rays the result would be returned as an array of 360 numbers, each one representing the corresponding distance to its nearest obstacle. We call this data structure the *all-around-array*, *AAA* for short. At elements without a value, respectively the maximum value initially set, there are no obstacles within the safe horizon. In Fig. 6.3, which illustrates a simplified AAA of 16 elements, we depict the the minimum distance to the nearest obstacle.

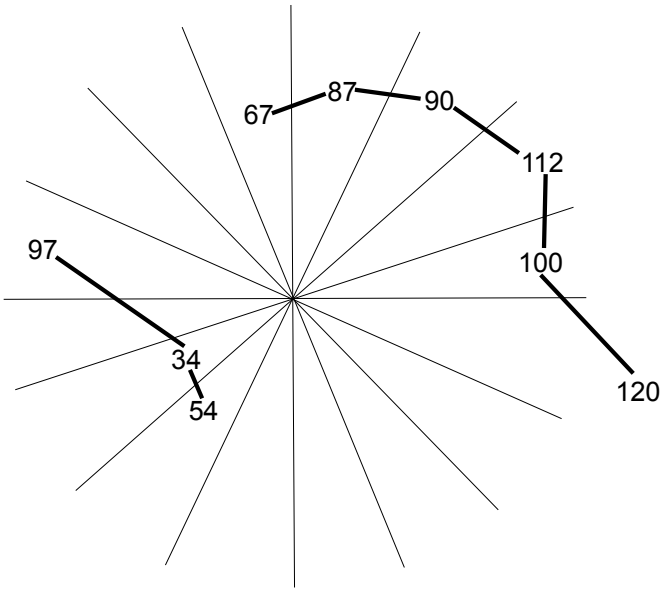


Figure 6.3: AAA-Values. Sectors without obstacles have no value respectively safe horizon r_{max} as default)

6.3 Weeding out non-relevant data (culling)

Only obstacles that are closer than r_{max} are considered relevant for collision avoidance. Since the total number of obstacles can be very large it is necessary to implement object culling in an efficient way. To avoid frequent distance calculations between the boat and every known object, the algorithm uses the following caching mechanism:

At the beginning all obstacles o_j are sorted based on their distance $d(o_j, S)$ to the starting point S. All initially relevant objects lie within $0 < d(o_j, S) < r_{max}$. After the boat has travelled a distance L the potentially interesting obstacles can be retrieved by querying the previous list for all objects for which

$$L - r_{max} < d(o_j, S) < L + r_{max} \tag{6.3}$$

This formula describes a doughnut with radius L and gauge $2r_{max}$ as shown in Fig. 6.4.

The resulting range of objects can be further reduced by taking the moving direction into consideration as indicated in the illustration. As the area of the doughnut grows with L, the sorting should be repeated relative to the current position S' whenever L exceeds a certain threshold.

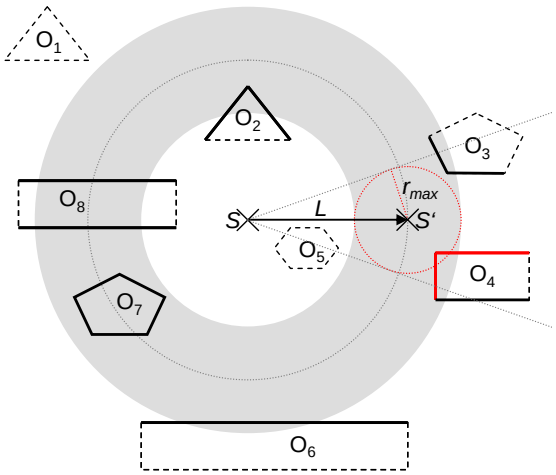


Figure 6.4: After travelling a distance of L, all now relevant objects lie within the grey area.

6.4 Sort and sweep algorithm

6.4.1 Description

For each sector only the distance to the closest obstacle is relevant. To calculate the minimum distance per sector we have to evaluate the obstacle polygons considered to be at a relevant distance from the boat. Sort-And-Sweep assumes that no two line segments intersect with each other. This is the case when the line segments are used as building blocks for simple, non-overlapping, non-intersecting polygons.

Sweep line algorithms are a major technique in algorithmic geometry. They led to a breakthrough in the computational complexity of geometric algorithms when Shamos & Hoey [1976] presented algorithms for line segment intersection in the plane. The technique itself may be traced to scanline algorithms of rendering in computer graphics. Wylie *et al.* [1967] introduced the scanline rendering technique.

The sort and sweep algorithm deals with obstacle data as described below. Fig. 6.5 furthermore illustrates the *sort and sweep* algorithm with the help of an example.

(a) Sort all points of the line segments radially, i.e. ascending according to their angular position relating to the boat's position. We call this the *points angular list*. Each point in this list maintains a link to its line segment. The list shall start with the point, which is geographically closest to the boat's position. For the case that this point is an ending point[1] of a segment, take the corresponding starting point of the segment in question as start point of the list.

(b) Take the first point of the *points angular list* and put the according line segment into the *list of current scan (LOCS)*, which is sorted by distance from the centre.

[1]The starting point is the first one if we sort them radially according to their angular position relating to the boat's position. The other point of the same segment is considered to be its ending point.

(c) Get the next point from the *points angular list*. This point can be either a beginning point of a new segment or an ending point of a segment. In the first case put the according segment into the LOCS. LOCS is maintained sorted by distance from the boat. In the other case the according segment is removed from LOCS. Possibly a new line segment becomes the element with minimum distance to the centre. Step (c) is repeated for each point of *points angular list*.

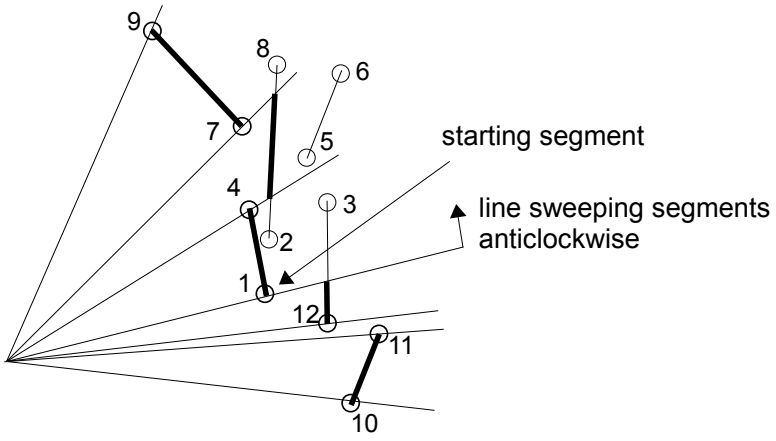


Figure 6.5: Example for the sort and sweep algorithm: the sweeping starts at the point with the smallest distance from the boat. Element changes of LOCS are indicated by small circles around the respective points. Not all points bring about a new minimum LOCS-element. At points 2, 3, 5, 6, and 8 the LOCS is changed but its minimal segment part stays in front.

6.4.2 Analysis

The non-intersection property guarantees that the number of points is the maximum number of executions of steps (c). LOCS maintains its sorting order during the sweep from one point to the next. Although the distance of each segment, especially of the closest segment, will change.

Step (a) needs $\mathcal{O}(n \log n)$ time (n the number of line segments), Step (b) $\mathcal{O}(n)$ and Step (c) needs for per manipulation of LOCS (insertion or deletion) $\mathcal{O}(\log n)$ and is repeated $2n - 1$ times. Therefore the whole algorithm is bound by $\mathcal{O}(n \log n)$ as well.

6.5 Minimal distance maintenance (flower algorithm)

In order to ensure a minimal safety distance r_{min} to any obstacle, additional evaluations are carried out. Just the modification of the polar diagram according to the AAA (see Fig. 6.2) is not sufficient. For instance, the boat would follow a route close to parallel along a straight obstacle line.

Looking into the direction of the route the distance might be larger than the allowed r_{min}, however the side-distance can still become less than that by following this route. Therefore, it is necessary to evaluate additionally whether a certain sector of AAA is allowed with respect to the other sectors. Assuming there is no obstacle within a radius r_{min} we can choose a test point in each sector with a distance of r_{min} from the current boat position. Only segments with a distance to the boat less than or equal to $2r_{min}$ are critical and need to be considered. A r_{min}-*violation* is detected, if one critical segment were to reach a distance less than r_{min} from one of the sector test points. In this case,the corresponding sector is marked forbidden and must not be chosen for the next route decision.

We call this the *flower algorithm* because the "blossoming" of r_{min} circles reminds us of petals. Its complexity is $\mathcal{O}(\#critical\ segments \cdot \#AAA\text{-}sectors)$.

Fig. 6.6 illustrates an example of the flower algorithm. The boat is considered to be in the centre of the 8-sectored circle with radius r_{min}. Only segments which are closer than $2r_{min}$ are critical and have to be evaluated for each sector-test-point (small knots on the r_{min}-circle around the boat). Note the east-north-east-sector: it has an AAA-value of more than $2r_{min}$ (dashed line), however its test-point gets into a less than r_{min} neighborhood. Therefore, this sector is marked forbidden. All sectors with its test-points closer than r_{min} to any obstacle line (red beacons) are marked forbidden and must not be entered in the boats next move. Sectors with a blue beacon (i.e. not cutting into any segment) are safe. For illustrative purposes, the beacons are drawn here only in the direction of the critical sectors.

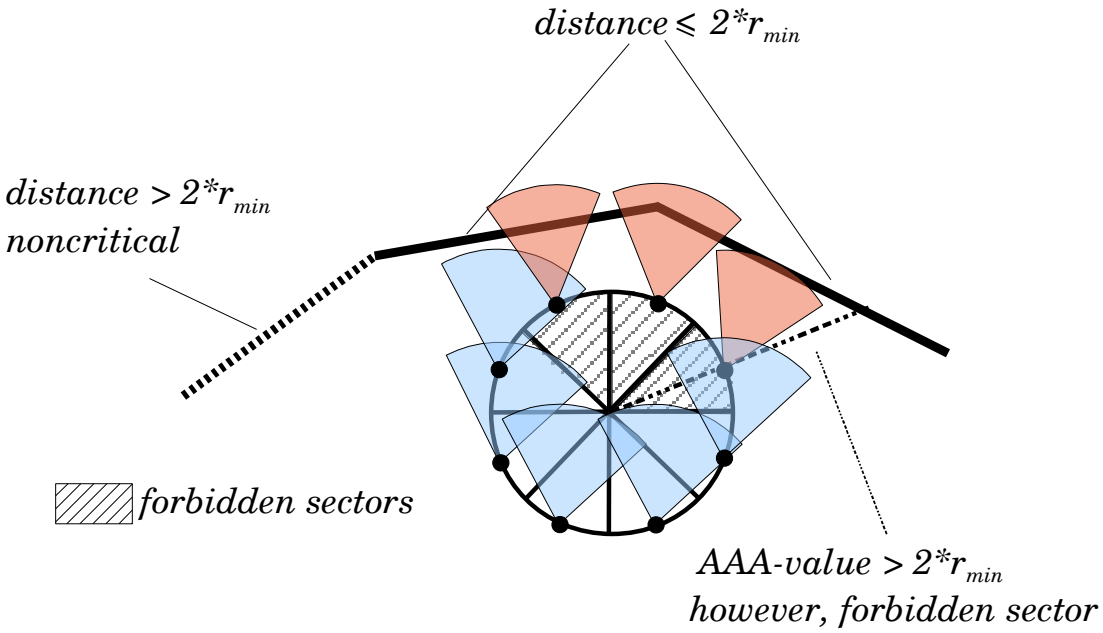


Figure 6.6: Example for mimimal distance mainainance (flower algorithm)

6.6 Experiments

6.6.1 Experimental setup

Experiments have been carried out using the same software implementation of the navigation algorithm which is used on *ASV Roboat*, but in a simulated environment. For the simulation a simple boat model is used, where rudder and sail movements are ignored and the simulated boat always follows the calculated heading with optimal speed. Fig. 6.7 shows the simplified polar diagram which is used in the simulation to describe the boat's behaviour. All directions are equally rated, except a 120 deg no go zone upwind which is set to 0.

For all experiments the overall distance between start and target destination was 1000 m. An obstacle, represented by a single line, was placed midway. The size of the obstacle was 50 m respectively 200 m. The values of the parameters are shown below:

- $r_{min} = 50$ m
- $r_{max} = 250$ m

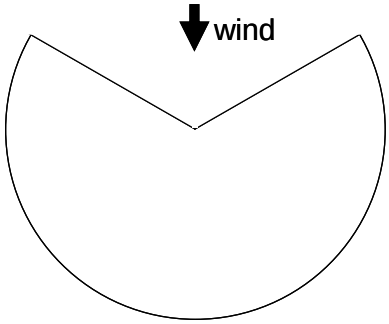


Figure 6.7: Simplified polar diagram

6.6.2 Results and discussion

Three scenarios with different courses were simulated. The simulated trajectories can be found in Fig. 6.8 for beam reach course, Fig. 6.9 for upwind course and Fig. 6.10 for downwind course respectively. All three scenarios where carried out with small (Fig. 6.8(a), 6.9(a), 6.10(a)) and large obstacles (Fig. 6.8(b), 6.9(b)–6.9(c), 6.10(b)–6.10(c)).

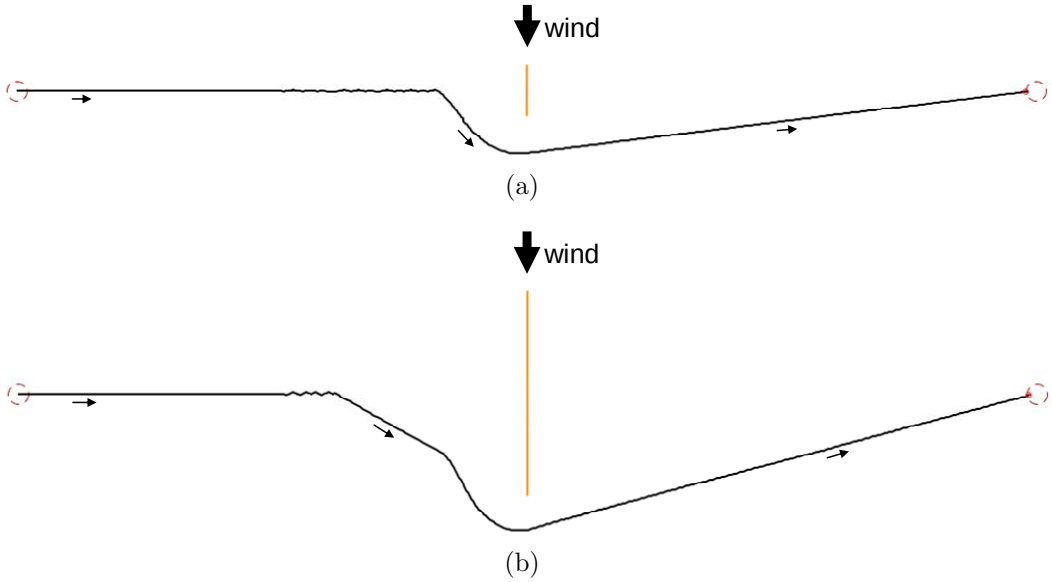


Figure 6.8: Simulation results in a beam reach

Small obstacles could be avoided easily on all simulated courses (Fig. 6.8(a), 6.9(a), 6.10(a)). When dealing with bigger obstacles the method shown starts

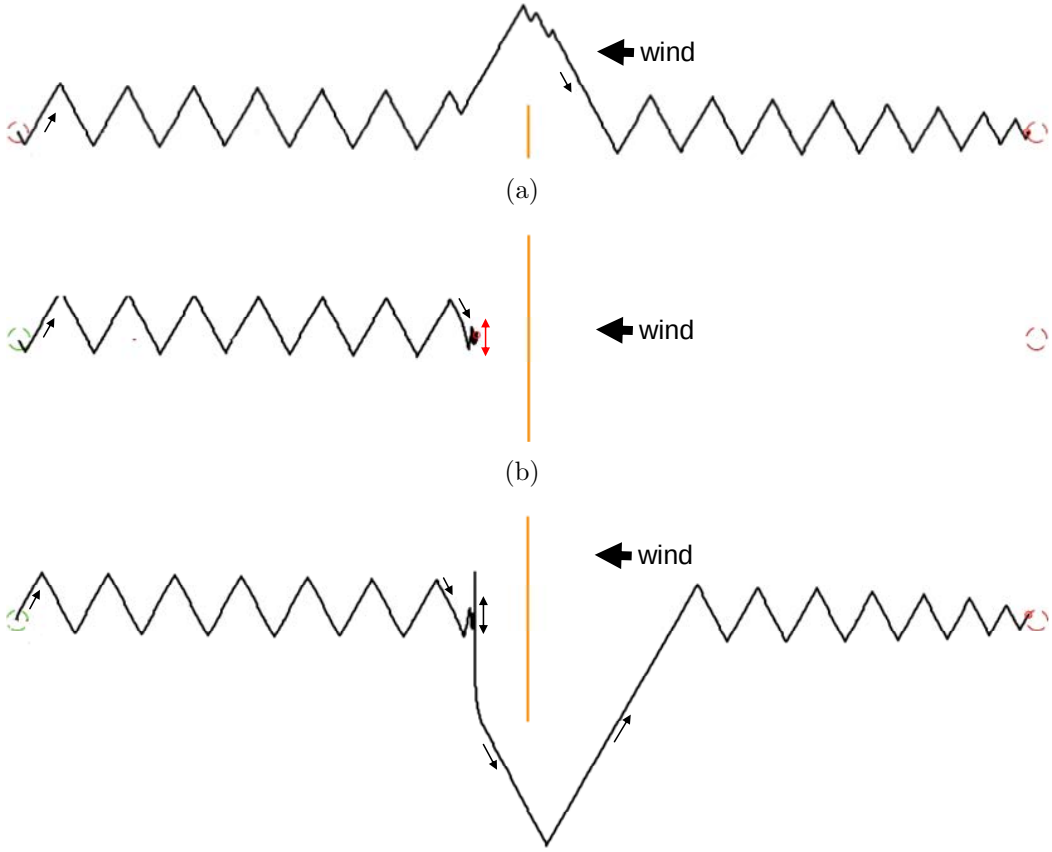


Figure 6.9: Simulation results upwind

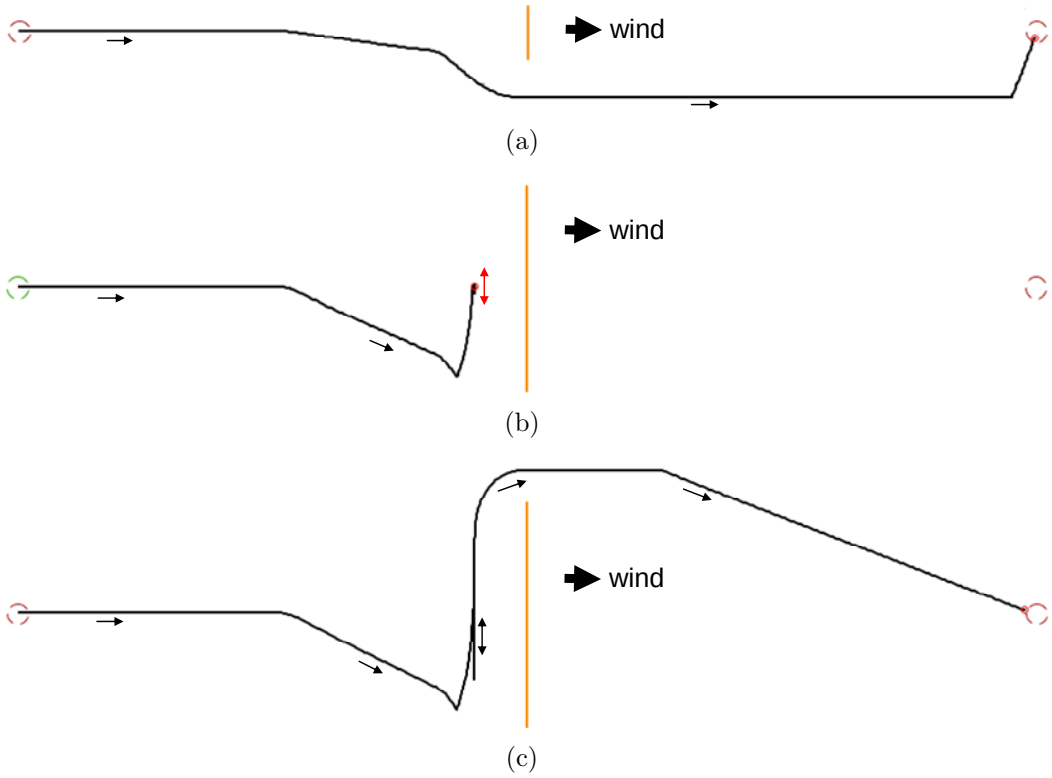


Figure 6.10: Simulation results downwind

to oscillate between two local optima v_t, which leads to the boat getting stuck in front of the obstacle (Fig. 6.9(b), 6.10(b)). This problem can be solved by detecting the oscillating behaviour, i.e. the boat does not advance to the target significantly after multiple manoeuvres whilst gradually tightening the hysteresis condition (Fig. 6.9(c), 6.10(c)). This means to double the beating parameter p_c (see Eq. 5.6) until the boat successfully circumnavigates the obstacle.

6.7 Summary

The method demonstrated is a novel reactive approach to obstacle avoidance for autonomous sailing boats. It is an extension to the short course routing method introduced in Chapter 5 which enables it to deal with obstacles in real-time. By contrast to the strategic long term routing (see Section 4.1.1) no planning is occurs here. Short course routing and obstacle avoidance techniques described here rely on locally measured data only and treat them in a reactive way.

Initial simulation results are promising. The algorithm enables an autonomous sailing boat to avoid obstacles by adapting the underlying polar diagram by putting a penalty on directions where obstacles are located within a certain range. Computer simulations with simple fixed obstacles in various sizes are promising. It turned out that for large obstacles an adaptive hysteresis condition has to be implemented in order to be able to steer around the obstacle.

Since the obstacle avoidance method is an extension to the short course routing which does not affect the interface between short course routing and manoeuvre execution layer (see Chapter 4) no further real world tests were necessary to validate the simulation data.

Future experiments will include moving obstacles, more complex and non-convex obstacles as well as arrangements of multiple obstacles. The results will be verified with real world experiments.

Chapter 7

Fuzzy logic actuator control and manoeuvre execution strategy

Many different systems are available that can assist the skipper steering a sailing boat. The most popular systems on the market are autopilots and wind vanes. Both systems can keep the boat on a predefined course. While the autopilot usually sails a predefined compass course, the wind vanes keep the boat on a certain course relative to the apparent wind. The systems control the rudder but have no influence on the sail position. The sheets still have to be adjusted manually. Therefore neither autopilots nor wind vanes can steer tack or jibe without human assistance. Automatic sail control is a more recent idea and hardly covered in scientific literature. So far, all sail control strategies have calculated a desired sail position just from locally measured apparent wind speed and direction. See Chapter 2 for a more detailed review.

The system presented in this work is able to control all manoeuvres of an autonomous sailing boat. A separate software module is responsible for short course routing (see Chapter 5) and delivers a desired direction according to the actual boat position and local wind conditions in real time. If the actual boat direction deviates from the desired direction, the system adjusts the rudder position in order to bring the boat back onto the desired course. In parallel, a second control system ensures that there is air flow in the sails in order to gain speed. By means of adjustments to the sheets, the boat's heeling is controlled. The sail

controller tries to keep the heeling at a calculated value dependent on the speed and direction of the apparent wind.

The boat must be able to maintain course, to make course corrections, and to carry out manoeuvres like tack and jibe. If such a manoeuvre is indicated due to significant changes of the desired boat direction, the control system ensures a smooth execution of the manoeuvres automatically.

The main aim of these control circuits is to imitate the behaviour of an experienced human sailor. Therefore it is not limited to a specific boat, but it is assumed to be applicable to any common type of sailing boat.

In common sailing practice, different persons are able to control rudder and sail independently without the need for communication. Therefore, in the present work, each of two independently working control loops is responsible for one of the actuators, rudder and sail. The rudder controller keeps or brings the boat on a predefined course determined by the routing software. The sail controller prevents capsizing and assures that there is enough air flow in the sails, which results in the boat's propulsion.

7.1 Fuzzy control system

Simple instructions for steering a sailing boat can be summarised in terms of a few rules. Fuzzy logic is particularly well suited to translating the natural-language rules of a control system into a computer program. Furthermore it allows smooth operation in an unpredictable, highly dynamic environment.

There are two main types of fuzzy inference systems used in control applications: the Mamdani type (Assilian & Mamdani [1974]) and the Sugeno type (Takagi & Sugeno [1985]). Due to the intuitive and interpretable nature of the rule base Mamdani type fuzzy inference systems (FIS) are used here to control rudder and sails. Mamdani type if-then rules can easily be understood by experts, experienced sailors in our case, even if they do not have any knowledge about control theory or fuzzy logic. Since the consequent part of a rule in a Sugeno type FIS is a function and not a word this interpretability is lost.

Determination of the fuzzy sets and rules is a key issue in fuzzy control and a difficult task, especially in an ever-changing environment, where experimental

data are hardly repeatable. A heuristic approach has been taken, where observation of the boats behaviour during remote-controlled sailing and discussions with experienced sailors led to an initial set of fuzzy terms and their membership functions. Like in many applications of fuzzy rule-based systems, also the fuzzy if-then rules have been obtained from human experts. Fine-tuning of the system has then been done during actual test runs.

Trapezoid fuzzy sets are used as inputs and singletons represent the output variables. Defuzzification is done by the centre of gravity of singletons method (CoGS). The crisp output values describe an amount of change and not an absolute value for the corresponding actuator. The frequency of execution of the FIS was identified by experiment and has to be adapted for a particular type of boat.

7.1.1 Rudder control circuit

Input data for the rudder control circuit are the current boat direction and the desired direction given by the short course routing system. The difference between these two gives the necessary course correction which enters directly into the fuzzy system as an input variable.

In order to avoid over steering, the angular velocity of the boat is used as an additional input variable.

Input variables

The inputs for the rudder FIS are *desired direction* and *turn*. The *desired direction* (deg) is given by the short course routing system, and the *turn* (deg/s) is the time derivative of the actual boat direction given by the compass. The fuzzy sets representing the linguistic terms of the variables are trapezoids (Fig. 7.1 and Fig. 7.2).

Output variable

The rudder control circuit's output is the change of the rudder position. The fuzzy variable *rudder change* (as a percentage of extreme positions) contains of five singletons representing the linguistic terms from *strong left* via *left*, *keep*, *right* to *strong right* (Fig. 7.3).

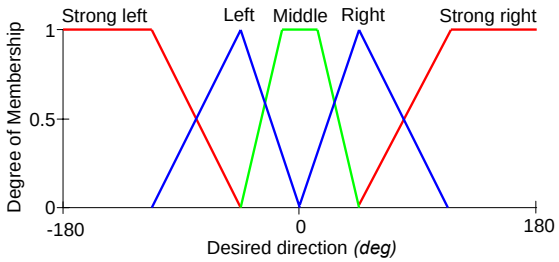


Figure 7.1: Fuzzy sets for input variable *desired direction*

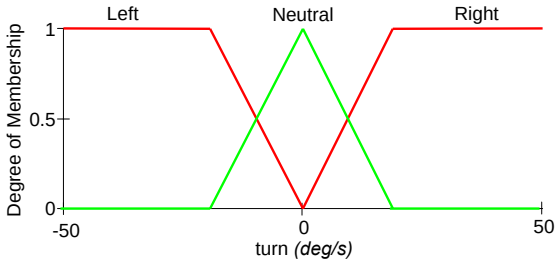


Figure 7.2: Fuzzy sets for input variable *turn*

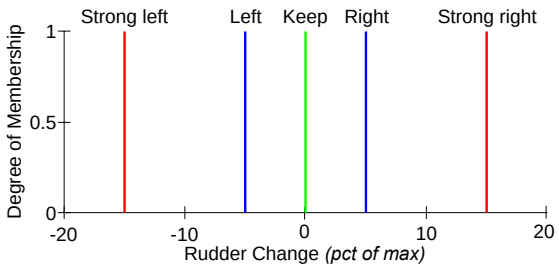


Figure 7.3: Singletons for output variable *rudder change*

Fuzzy rules

The rule base of the rudder FIS contains 15 rules in the form

IF *desired direction* IS x AND *turn* IS y THEN *rudder change* IS z

Table 7.1 shows the concrete parameters of the rules. Defuzzification is done by evaluating the centre of gravity of singletons (CoGS).

Rudder change		**Turn**		
		Left	**Neutral**	**Right**
Desired dir.	**Strong left**	Left	Strong left	Strong left
	Left	Keep	Left	Strong left
	Middle	Right	Keep	Left
	Right	Strong right	Right	Keep
	Strong right	Strong right	Strong right	Right

Table 7.1: Fuzzy rules for rudder FIS

7.1.2 Sail control circuit

The inputs for the sail control circuit are the heeling of the boat and direction and speed of the apparent wind. The sail FIS calculates direction and amount of necessary adjustment of the sail winch. The aim of the sail control circuit is to keep the boat's heeling at an optimum according to the actual wind conditions.

Calculation of desired heeling

The basic idea is to describe the following relationship between desired heeling and apparent wind.

- The higher the wind speed, the higher the desired heeling.
- The more the boat moves towards downwind the smaller the desired heeling.

$$h = max\left(0, (h_{max} - k \cdot |\alpha|) \cdot \frac{min\,(v, v_{max})}{v_{max}}\right) \tag{7.1}$$

Equation 7.1 describes the relationship in a formal way. The desired heeling h is a function of speed v and direction α of the apparent wind (Fig. 7.4). The constants k, h_{max}, and v_{max} are boat specific and determined by experiments. h_{max} describes the maximum desired heeling for apparent wind speed of v_{max} or above. The graphical representation of the heeling function is illustrated in Fig. 7.4.

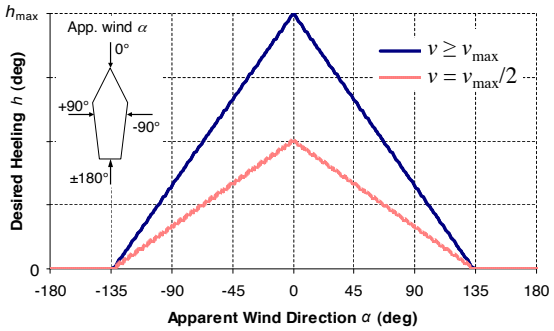


Figure 7.4: Desired heeling function

Input variable

The input for the sail FIS is the variable *heeling* (deg), which is the difference between the desired heeling and the actual heeling of the boat. Three trapezoidal fuzzy sets describe the linguistic terms *too low*, *optimal*, and *too high* (Fig. 7.5).

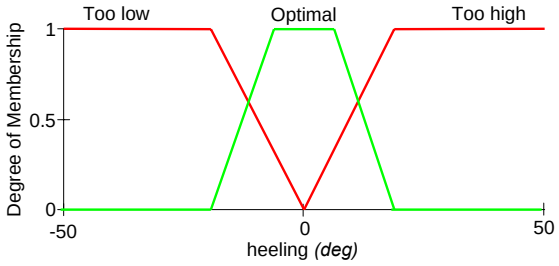


Figure 7.5: Fuzzy sets for input variable *heeling*

Output variable

The sail control circuit's output is the change in sheet winch position, which influences the sail position directly. The fuzzy variable *sail change* (as a percentage

of extreme positions) contains three singletons representing the linguistic terms *tighten*, *keep*, and *ease off* (Fig. 7.6).

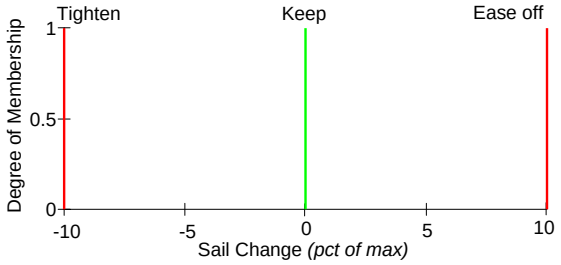


Figure 7.6: Singletons for output variable *sail change*

Fuzzy rules

The rule base of the sail FIS contains the three IF-THEN-rules:

- IF *heeling* IS *too low* THEN *tighten sheets*.
- IF *heeling* IS *optimal* THEN *keep sheets*.
- IF *heeling* IS *too high* THEN *ease off sheets*.

7.1.3 Manoeuvres

It is assumed that the weather routing system only provides courses which are navigable efficiently. In order to sail every given course, the system must be able to

- Maintain a given course
- Execute a tack and
- Execute a jibe.

Maintain a given course

Parallel execution of the sail and rudder control circuits guarantee to keep the boat on course by immediately adjusting the rudder in the case of course deviation. The sails are adjusted to maintain propulsion.

Execute a tack

If the direction given by the routing system changes significantly a tack may be necessary. The following procedure happens in this case (Fig. 7.7):

1. The sailing boat turns its bow towards the wind. Due to the change in the angle of the apparent wind, heeling decreases. Therefore, the sail control system tightens the sheets.

2. The bow turns through the wind so that the wind changes to the other side of the boat. Heeling increases again. Therefore the sail controller eases off the sheets.

3. The boat reaches its new course.

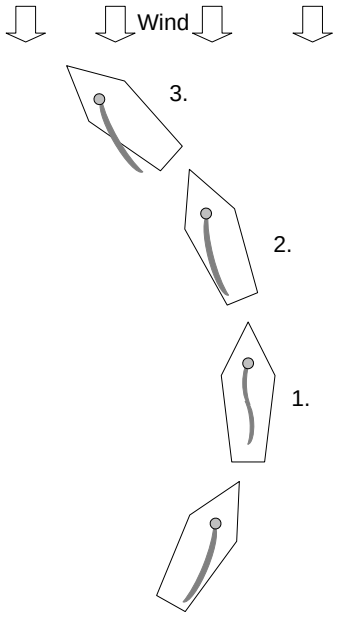


Figure 7.7: Tack execution

The tack is implicitly executed by the two fuzzy systems, a new given desired direction causes this type of manoeuvre.

Execute a jibe

As for human sailors the jibe is a bit more difficult than the tack and optimal timing is more important. In addition to the parallel execution of rudder and sail FIS a special rule has to be applied during the jibe. The following procedure happens in this case (Fig. 7.8):

1. The sailing boat turns its stern towards the wind. On exact downwind direction the desired heeling is 0. Therefore the sail is completely eased off.

2. Special Rule for the jibe: If the stern has turned through the wind significantly (hysteresis), but the sail is still on the windward side, the sail gets tightened temporarily in order to move to the leeward side. The hysteresis condition avoids permanent changes of sail position on a downwind course with minimal changes of wind or boat direction.

3. The boat reaches its new course.

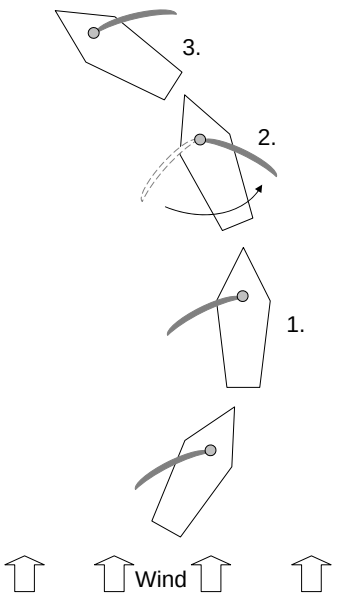


Figure 7.8: Jibe execution

7.2 Experiments on manoeuvre execution

7.2.1 Experimental setup

Several test runs have been carried out to demonstrate feasibility and suitability of the presented approach for execution of sailing manoeuvres. The data refers to the final test run prior to the Microtransat competition in France, where the wind conditions were within the operational range of the demonstration boat *Roboat I* (see Section 3.1 for details). Tack and jibe are analysed in detail. Fig. 7.9 and Fig. 7.10 show the process of manoeuvre execution. Log data at a 2 s interval are illustrated. The boat drawings present boat heading, sail position, rudder position, and apparent wind direction. Additionally, heeling and time lapse are stated. The wind arrows indicate the direction the wind is blowing to.

7.2.2 Results and discussion

Tacking

The tack starts on an initial course of 207 deg and ends on a target direction of 114 deg. The true wind direction is around 344 deg and its average speed is 1.2 m/s. The average boat speed in the beginning and the end of the manoeuvre is about 0.4 m/s and decreases slightly after 10 s, when the boat is pointing directly into the wind. The boat executes the complete tack successfully within 16 s and a trajectory of 6 m. The detailed process flow is illustrated in Fig. 7.9 and described in Table 7.2.

Jibing

The jibe starts on an initial course of 312 deg and ends on a target direction of 40 deg. The true wind direction is around 326 deg and its average speed is 1.2 m/s. The average boat speed is 0.6 m/s and does not change significantly throughout the entire manoeuvre.

The boat executes the complete jibe successfully within 16 s and a trajectory of 10 m. The detailed process flow is illustrated in Fig. 7.10 and described in Table 7.3.

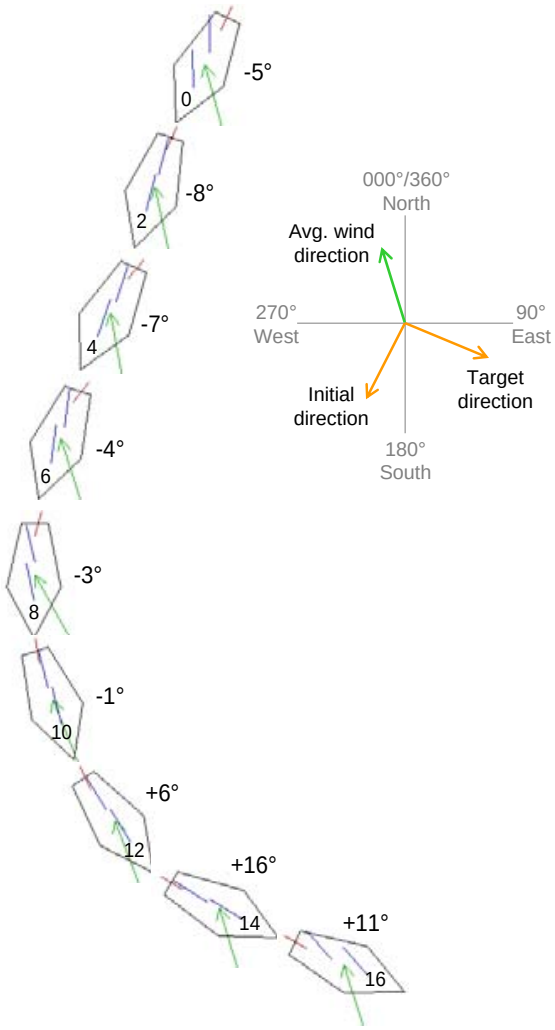


Figure 7.9: Tack from test run in 2 s time interval: rudder and sail position, apparent wind direction, heeling in deg, time from beginning in seconds.

Time in *s*	Process
0, 2	The boat is sailing close hauled on a port tack (initial course).
4	The boat receives a new direction by the routing system and initiates the tack by moving the rudder.
6	The boat begins to turn and rudder inclination increases.
8	Due to the turn of the boat the rudder inclination does not further increase, even though the boat is still far from its desired direction.
10	The boat is pointed directly into the wind. Heeling decreases below the desired value due to a lack of lateral wind force. Thus the sail control circuit tightens the sheets in an appempt to reach the desired heeling. To avoid over steering the rudder is already back in the central position.
12	Due to mass inertia the boat keeps turning towards desired direction.
14	Lateral wind forces increase. Hence heeling increases above desired heeling.
16	New desired direction is reached. Sheets are eased off in order to reach the desired heeling.

Table 7.2: Tacking timeline (according to Fig. 7.9)

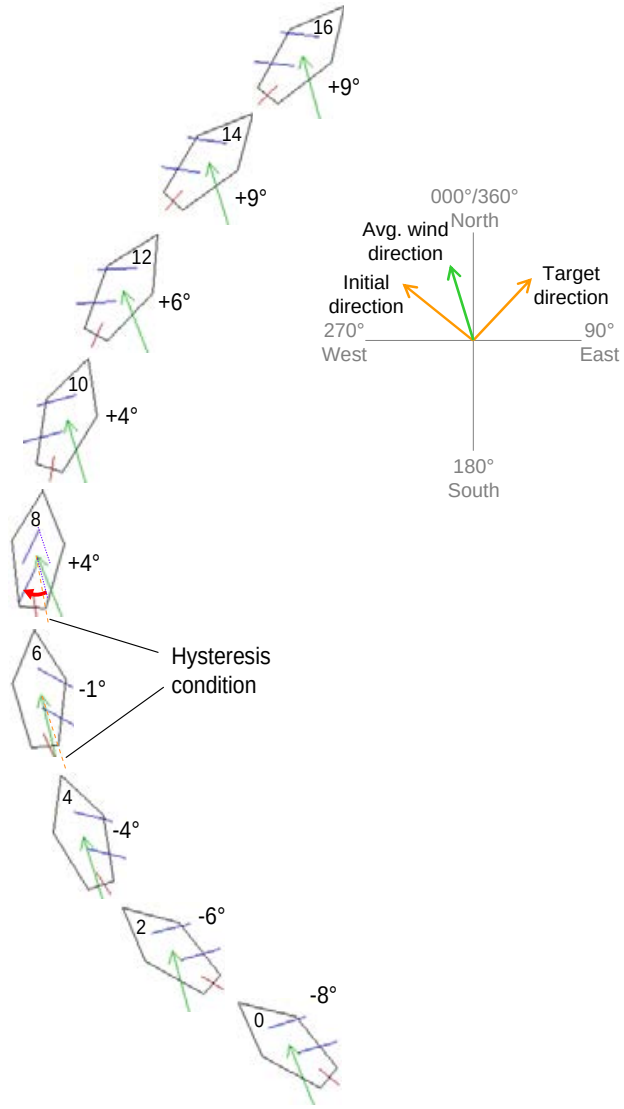


Figure 7.10: Jibe from test run in 2 s time interval: rudder and sail position, apparent wind direction, heeling in deg, time from beginning in seconds.

Time in *s*	Process
0	The boat is sailing a broad reach on a port tack (initial course).
2	The boat receives a new direction by the routing system and initiates the tack by moving the rudder.
4	The boat begins to turn and rudder inclination increases.
6	The boat's stern has already turned through the wind, although the sails are still on starboard side, because the hysteresis condition is not yet fulfilled
8	Now the stern has turned significantly through the wind and the hysteresis condition is fulfilled. Therefore the sheets get tightened temporarily in order to move the sails to leeward side.
10	Now the boat is sailing a broad reach on the starboard tack. Sails are eased off completely again, because desired heeling is low on a broad reach course. The rudder inclination decreases, because target direction is almost reached.
12, 14	The rudder is back near middle position. Due to mass inertia the boat keeps turning towards desired direction.
16	New desired direction is reached. Sheets are eased off in order to reach desired heeling.

Table 7.3: Jibeing timeline (according to Fig. 7.10)

7.3 Experiments on course keeping

7.3.1 Experimental setup

Beside execution of tack and jibe, course keeping is another key functionality of the manoeuvre execution layer. Aim of this experiment is to validate the following hypotheses:

1. The proposed FL system for rudder control enables an autonomous sailing boat to sail a given heading precisely.
2. The course deviation is not influenced significantly by the point of sail.
3. The course deviation is not influenced significantly by the wind speed.

The analysis has been carried out with the robotic sailing boat prototype ASV Roboat and is based on a set of log data recorded during the Endurance Race at the World Robotic Sailing Championship on Lake Ontario, Kingston, Canada in 2010. The ASV Roboat had to sail as many laps as possible on a triangular course within 8 hours.

All sensor values as well as calculated values from the control system on board have been recorded once per second. 25536 lines of log data therefore describe approximately 7 hours of continuous sailing. The triangular course was sailed 28 times which corresponds to a completed distance of 28 km. Fig. 7.11 show the trajectory of the analysed run . The triangular shape of the course guarantees to have a sufficient number of log lines for every point of sail. Furthermore various wind speeds from 0.06 m/s to 9.33 m/s have been recorded (Fig. 7.12). This allows a detailed analysis of different wind speeds within the operational range of the boat.

For this study we compare the desired heading which is determined by the short course routing system with the actual boat heading measured with a Maretron SSC200 compass on board. Especially on upwind courses the desired heading is influenced by wind shifts. Therefore, this analysis covers both steering a constant compass course and steering a constant point of sail.

For the statistical analysis of course keeping accuracy, log data during tacking and jibing need to be omitted. The start of a manoeuvre is indicated through

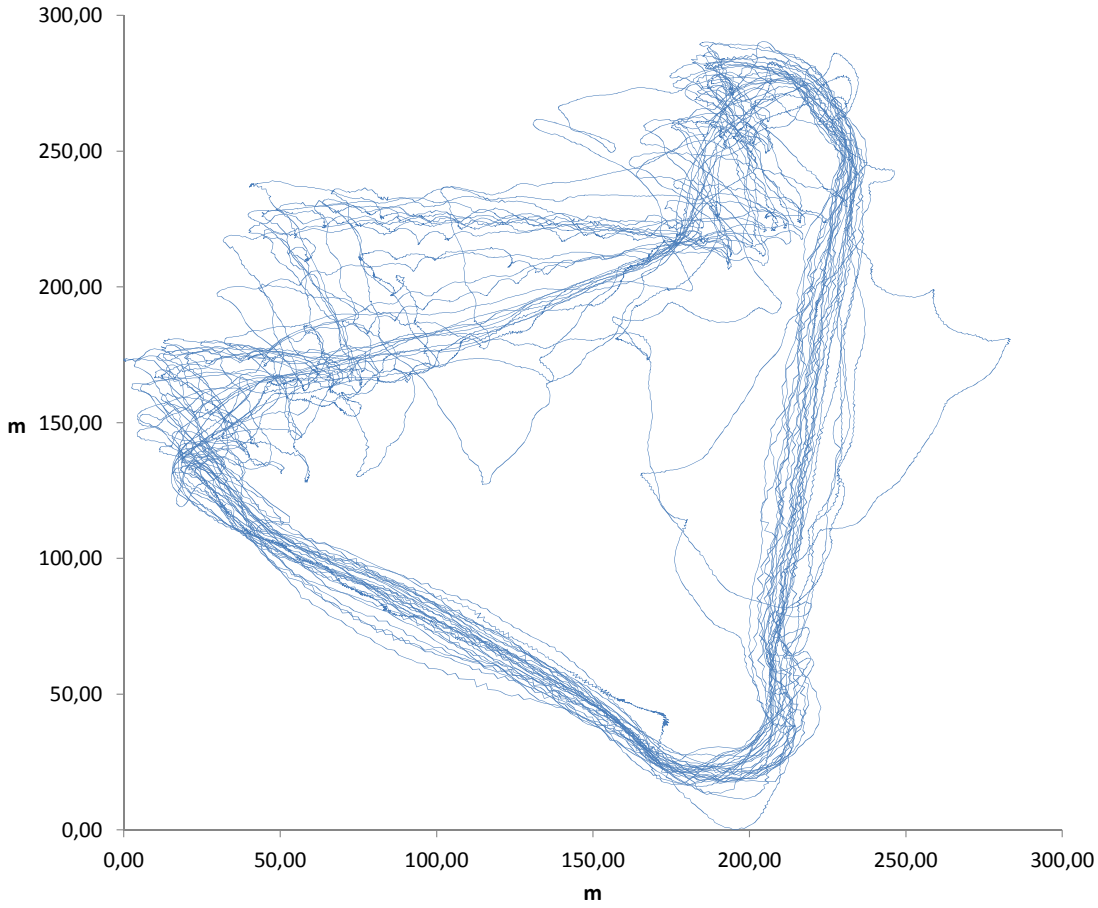


Figure 7.11: Trajectory from analysed test run on Lake Ontario, Canada in 2010

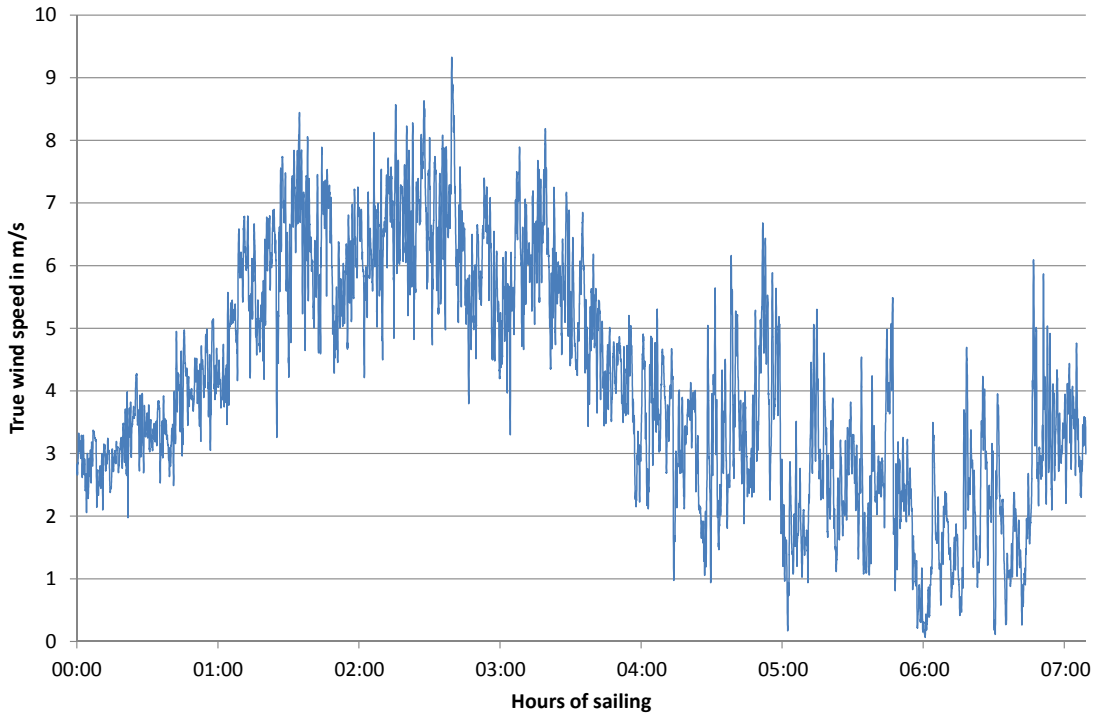


Figure 7.12: True wind speed during the analysed test run on Lake Ontario, Canada in 2010

a sudden significant change in the desired direction (< 10 deg). The manoeuvre is considered to be finished, when the boat is back on the desired heading again (heading error < 1 deg again). Furthermore, there are a few short sequences, where the boat was under manual control in order to avoid a collision. The corresponding log records have to be omitted too. A set of 11475 log lines remain for analysis.

7.3.2 Results and discussion

General course keeping performance

Fig. 7.13 shows the normal distribution of the sample data.

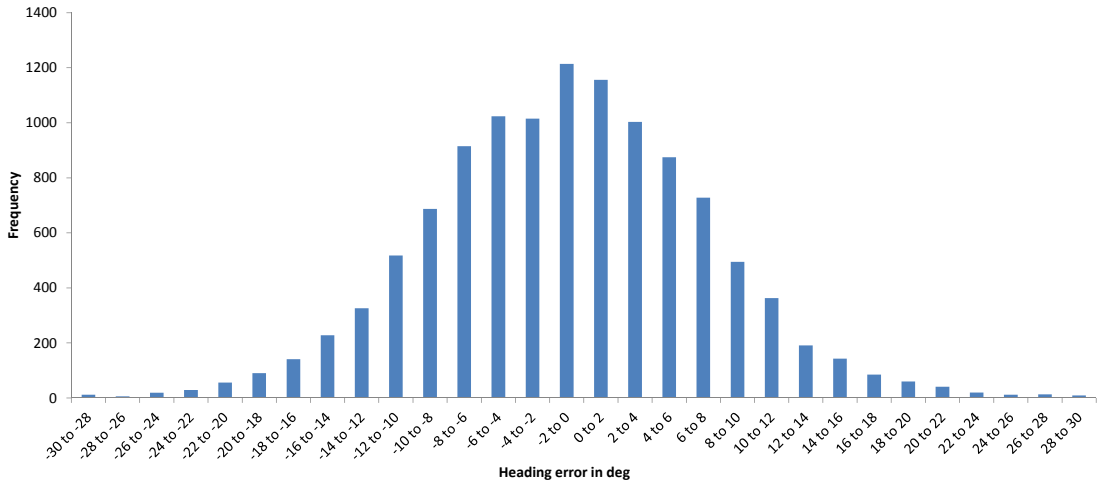


Figure 7.13: Normal distribution of heading error

Observations of several test runs, including the one which is analysed here, have shown that the boat is able to follow a given course precisely. Table 7.4 shows the results of a statistical analysis of the entire sample. It is obvious that the mean is close to 0 as both boat design and control system design are in a symmetric way. The mean error of -0.94 deg is furthermore within the inaccuracy of the sensor which is up to ± 1 deg according to the product data sheet of the compass. The more meaningful values relating to the course keeping quality are the standard deviation of 8.07 deg and the maximum error of approximately ± 30 deg. These values are clearly acceptable considering the highly dynamic

environment with ever-changing wind and waves acting on the boat and therefore validate the observations statistically.

Observations	11475
Mean	−0.94 deg
Standard deviation	8.07 deg
Minimum	−29.91 deg
Maximum	29.45 deg

Table 7.4: Statistics of heading error (total sample)

A detailed analysis on the impact of point of sail and wind speed on the course keeping quality is provided in the subsequent sections. A two-sided F-test is used, because this is appropriate to test whether the standard deviation of one set of data is different from another set of data (Morgan [2006]). In our case the total sample is subdivided into several clusters, each representing a certain range of wind speed respectively a particular point of sail. Then the set of heading error data from each cluster is tested against a set of all other heading errors.

Influence of point of sail on course keeping

The sample is subdivided into several clusters, each representing a particular point of sail. Table 7.5 shows the ranges for the analysed points of sail and the number of log data for each cluster.

Point of sail	Range of wind direction	Observations
Close hauled	0 deg to 50 deg	6523
Beam reach	50 deg to 100 deg	8085
Broad reach	100 deg to 150 deg	9505
Running	100 deg to 180 deg	10312

Table 7.5: Clusters according to the point of sail. The wind direction is given as a relative angle to the boat's heading (0 deg means the boat is facing into the wind and 180 deg means the wind is coming directly from behind the boat). Port and starboard wind is not considered separately.

We tested the following hypothesis:

- H_0: The course deviation is not influenced by the point of sail.
- H_1: The course deviation is influenced by the point of sail.

The results were calculated in R[1] using the var.test command. The comparison of any particular point of sail to all other points of sail reveals no statistically significant difference in the course keeping ability of the boat at the 95 % confidence level. Table 7.6 shows, that the given set of data does not reject the null hypothesis for all selected points of sail.

	F	CI ($\alpha = 0.05$)	H_0
Close hauled	0.788	0.748 to 0.830	not rejected
Beam reach	1.547	1.462 to 1.638	not rejected
Broad reach	0.930	0.869 to 0.997	not rejected
Running	0.674	0.620 to 0.736	not rejected

Table 7.6: Analysis of point of sail on course keeping quality.

Influence of wind speed on course keeping

The sample is subdivided into several clusters, each representing a particular range of wind speed according to the Beaufort scale. Table 7.7 shows the ranges for the analysed wind speeds and the number of log data for each cluster.

Bft.	Description	Range of wind speed	Observations
0,1	**Calm and light air**	0 m/s to 1.5 m/s	1147
2	**Light breeze**	1.5 m/s to 3.3 m/s	3864
3	**Gentle breeze**	3.3 m/s to 5.4 m/s	3853
4	**Moderate breeze**	5.4 m/s to 8 m/s	2509
5	**Fresh breeze**	8 m/s to 10.4 m/s	102

Table 7.7: Clusters according to the Beaufort scale for wind speed.

[1] `http://www.r-project.org`

We tested the following hypothesis:

- H_0: The course deviation is not influenced by the wind speed.
- H_1: The course deviation is influenced by the wind speed.

The results were calculated in R using the var.test command. The comparison of any of the individually evaluated wind speed range to all other wind speeds reveals no statistically significant difference in the course keeping ability of the boat at the 95 % confidence level. Table 7.8 shows, that the given set of data does not reject the null hypothesis for all selected points of sail.

	F	CI ($\alpha = 0.05$)	H_0
Calm and light air	2.524	2.319 to 2.756	not rejected
Light breeze	0.933	0.883 to 0.985	not rejected
Gentle breeze	0.744	0.705 to 0.786	not rejected
Moderate breeze	0.780	0.734 to 0.832	not rejected
Fresh breeze	0.882	0.681 to 1.188	not rejected

Table 7.8: Analysis of impact of wind speed on course keeping quality.

7.4 Summary

Sailing experts can explain basic sailing skills by rules about how to steer sails and rudder according to direction of target and wind. The presented solution describes how to transform the sailor's knowledge into Mamdani type fuzzy inference systems.

It can be summarized that an autonomous sailing boat can be effectively controlled by two independent Mamdani type fuzzy inference systems: one for the rudder, the other for the sails. This reflects common sailing practice where different people act more or less independently on rudder and sails respectively. As optimum air flow in the sails is hard to measure, this novel approach to sail adjustment controls towards an optimum heel angle instead of a concrete sail position. This heel angle depends on the speed and direction of the apparent wind.

This system basically allows the boat to maintain a predefined course and ensures a suitable sail position. Changes in the desired direction may lead to manoeuvres like tack or jibe. Tacking is executed implicitly by the fuzzy control circuits. Only the jibe requires an additional rule in order to move the sails to leeward side.

Experiments on an autonomous sailing boat have been carried out to demonstrate feasibility and suitability of the presented approach. Log files from actual sailing trips show a steering behaviour as expected by sailing experts. The boat executes the complete tack or jibe successfully within about 15 s. The test runs did not show any conflict between the two fuzzy inference systems executed in parallel.

Statistical analysis has shown, that the proposed FL system for rudder control enables an autonomous sailing boat to sail a given heading precisely. Furthermore, no statistically significant impact on the course keeping ability of the boat could be observed through changes in point of sail or in wind speed.

Chapter 8

Discussion and further work

In the following section it is evaluated how the results of the presented research have met the original aims. Each of the research questions stated in Section 1.4 is analysed individually. Results as well as their limitations are discussed.

8.1 Sailing boat routing

> *How can an optimum route to any given way point be determined considering locally measured wind data and wind forecasts? What are the differences between short course and long term routing?*

The initial parameters for route planning are the boat's current position and the coordinates of the destination, which the boat is intended to reach as efficiently as possible. The most significant influencing factors in routing a sailing boat are the wind conditions and any obstacles present. For this study, routing process for large distances was divided into two stages. Firstly, strategic long-term routing (Section 4.1.1) calculates an approximate route, taking into account weather forecasts and already known fixed obstacles such as land masses, shoals or areas that may not be entered for legal reasons. The result of this strategic long-term routing is a series of intermediate way points that divide the overall route into short sections. As the boat progresses, each successive way point is passed to the next routing level - short-course routing (Section 4.1.2).

8.1.1 Long term routing

As result of a joint research project of Johannes Langbein, Thom Frühwirth (both Ulm University, Germany), and the author a strategy for long-term routing of autonomous sailing boats has been proposed (Langbein *et al.* [2011]). It is based on the A*-algorithm, which incorporates changing weather conditions by dynamically adapting the underlying routing graph (see Appendix A). The algorithm has been implemented in the declarative rule-based programming language Constraint Handling Rules (CHR) (Frühwirth [2009]). It works with real-life wind forecasts, takes individual performance parameters of the sailing boat into account, and provides a graphical user interface.

A comparison with existing commercial applications Sailplanner (Sailport AB [2011]) and SailFast (SailFast LLC [2011]) yields considerably shorter computation times for this implementation. The routes computed by our algorithm and SailFast are almost identical, while the route computed by Sailplanner is somewhat different. There are two reasons for this difference: Firstly, Sailplanner uses a different polar diagram, which could not be changed in the demo version available to the authors. Secondly, Sailplanner was run with wind data from WeatherTech, as it does not allow the import of GRIB files. The presented algorithm and SailFast were both run with the same GRIB file (WMO94) from `saildocs.com` since the data from WeatherTech is not freely available.

8.1.2 Short course routing

Short course routing on autonomous sailing boats in real world conditions can be implemented in the first instance with the aim of imitating the behaviour of a human sailor. In the present work, a technique is presented to determine suitable boat headings in order to reach any target. The method works without knowledge of future weather conditions. This is advantageous especially for short distances and short periods of time, where no accurate weather forecasts are available. This method is relatively simple and easy to implement, even on an embedded system. A parameter defines the width of a potential beating area. This beating parameter can be used to ensure the boat stays within a safe area.

Simulations have shown that the routing strategy does not rely on the knowledge of the specific boat behaviour (polar speed diagram). The velocity made good is an important variable to be globally optimised within the routing strategy. However, it can be shown that simply continuously maximising the velocity made good does lead to sub-optimal results. The best results are obtained with a simplified polar diagram, which only defines the efficiently navigable range in terms of angles between boat and wind. The simplified polar diagram forces the boat to take the direct straight route even in situations where the boat polar diagram suggests a different direction that is currently better but leads to a worse overall performance.

The parameters needed for the calculation of the desired boat heading are:

- target position
- current boat position
- current boat heading
- true wind direction

Tests of the method on an autonomous sailing boat show its strength in dealing with a highly dynamic environment. The algorithm reacts in real-time to changing wind similar to a human sailor sailor. A weakness of the current implementation is, that at reasonable distance to the target even small changes in wind direction will cause a tack or jibe.

The characteristic behaviour of the demonstration boat has been determined. By comparing an experimental run to a computer simulation the boat specific relationship between wind, boat velocity, and leeway has been determined. If leeway is considered in the computer model of the sail boat, the actual data log from the experiment and the simulated course match well.

The routing strategy does not yet take into consideration obstacles like land masses or extreme weather phenomena. These aspects can potentially be combined with long-term routing methods on the one hand and an extension to this algorithm on the other (Chapter 6).

8.2 Collision avoidance

How can a robotic sailing boat navigate safely and efficiently around obstacles?

To reliably avoid collisions, potential sources of danger must be detected and then responded to appropriately. The present study has given particular consideration to the latter, that is to strategies for safe avoidance, without involving excessive detours.

Since avoidance is regarded as an extension to the routing process, it is processed analogously, on two parallel levels: (1) planned in the range of long-term routing, and (2) reactive methods within a specified range as part of short-course routing.

An algorithm for the long-term routing functionality has been jointly developed with scientists at Ulm University in Germany (Appendix A). This implementation considers fixed and previously known obstacles when planning a rough route for long distances. A further qualitative evaluation of the calculated routes has not been done so far, as no test runs over reasonable distances have been carried out. The system can be used not only on robots, but also on conventional sailing boats.

Where unplannable obstacles arise in the boat's immediate surroundings, a reactive process is used. The method presented represents an extension of the short-course routing algorithm that was also developed as part of the present study. The routing algorithm indicates the direction that offers the optimal approach to the destination in view of local wind conditions (maximum VMG). The basis for this calculation is the boat's specific polar diagramme. When obstacles are detected nearby, the polar diagramme is dynamically modified in such a way that headings in the direction of obstacles are evaluated as poorer. In an extreme case - where the obstacle is dangerously close - the appropriate direction is marked as completely un-navigable and eliminated as a routing option.

The computer simulation showed good results with simple obstacles. They were safely and efficiently bypassed. A comparable avoidance strategy would be expected of a human sailor if no information such as nautical charts or weather forecasts were available to allow him or her to make advanced plans. Tests with

moving or non-convex obstacles have not yet been carried out.

The method responds quickly and smoothly but could be substantially improved if additional information about the obstacles could be incorporated. This would be for example the course and speed of nearby ships. Such data could be received via an AIS receiver. Additionally, depending on the type of obstacle, different safety distances could be applied. It would for example be possible to sail closer to precisely measured, stationary objects than to moving obstacles or those that are detected with imprecise sensors.

8.3 Sail and rudder control

How can human sailors' navigation knowledge and experience be implemented within a computer program? Is fuzzy logic (FL) an appropriate method of keeping an autonomous sailing boat on course and of carrying out sailing manoeuvres? Do the methods work properly on differently sized boats?

Simple instructions for steering a sailing boat can be summarised in terms of a few rules. Fuzzy logic is particularly well suited to translating the natural-language rules of a control system into a computer program.

Three fuzzy variables (13 fuzzy sets in total) and 15 simple rules that a sailor would be able to understand are sufficient to control the rudder. This allows the boat to:

- maintain course
- make small corrections of course (e.g. if the wind should change direction) and
- to steer itself by means of tacking and jibing.

The sail is controlled by a second, standalone control loop. Optimal heeling is calculated from the locally measured wind conditions and the course in relation to the wind. A fuzzy interference system (FIS) then controls the boat on this

heeling. This FIS requires only two fuzzy variables (six fuzzy sets altogether) and three rules.

Observations of several test runs and statistical analysis of log data have shown, that the proposed FL system for rudder control enables an autonomous sailing boat to sail a given heading precisely make small corrections of course if indicated.

The mean heading error of -0.94 deg is within the inaccuracy of the compass which is up to ± 1 deg. The more meaningful values relating to the course keeping quality are the standard deviation of 8.07 deg and the maximum error of approximately ± 30 deg. These values are clearly acceptable considering the highly dynamic environment with ever-changing wind and waves acting on the boat and therefore validate the observations statistically.

Furthermore, no statistically significant impact on the course keeping ability of the boat could be observed through changes in point of sail or in wind speed. Neither the analysis of several wind speed ranges nor of different points of sail did reveal any statistically significant difference in the course keeping ability of the boat at the 95 % confidence level.

Real-world tests of the rudder and sail control systems have been performed with two prototype sail boats that differed significantly from one another in characteristics such as size, weight and rig (Sections 3.1–3.2).

The greatest effects caused by the differences between the two prototype boats were expected in the area of rudder and sail trim. Without altering the configuration relative to the 1.38 m long Roboat I, the 3.72 m long ASV-Roboat oversteered considerably by tacking and jibing and was inclined on downwind courses to oscillate. The reason for this was the considerably higher inertia coupled with a slower rudder drive. It turned out that for the most part, the change to the larger boat required only that the fuzzy system be artificially slowed down. While the sail and rudder positions were recalculated and adjusted every 50 ms on the small boat, on the larger it was sufficient to do so only every 300 ms.

With wind forces above about 5 Bft and unfavourable waves, oversteering or understeering may still occur after manoeuvres. By implementing a state machine it was possible to treat the massive changes in direction that occur during manoeuvres in the rudder control system separately. A state machine for

tacking and jibing has also been used on the following boats for this purpose: Däumling (University of Lübeck, Germany), Avalon (Swiss Federal Institute of Technology Zurich, Switzerland) and IBoat (ISAE, France) (Briere [2008]; Burnie [2010]; Giger *et al.* [2009]).

The matter of whether the methods would scale so well for even larger boats is assumed but has yet not been proven. A computer simulation using a generalised kinematic boat model could provide information on this. The steering methods were not tested on multihulls or on dinghies, nor with wing sails.

8.4 Communication

How can a reliable data link between boat and shore be guaranteed for the development and future applications of autonomous sailing boats?

Several test runs have shown that real-time communication for monitoring and remote control is of importance for safety especially in the field of vision close to shore. In case of emergency immediate manual control has to be possible. On the other hand, for long-term missions over huge distances the main focus is on global network coverage and reliable transmission of a few essential values. Higher transmission latency can be accepted.

A three-stage communication system for autonomous sailing boats has been designed, implemented and tested successfully. The advantages and disadvantages of the individual stages have been illuminated with regard to availability, costs, bandwidth, and real-time abilities. Requirements during development of autonomous sailing boats as well as their applications have been considered separately.

Based on the experiments carried out, the presented concept turned out to be appropriate. Long-term test runs have to be done in order to get more valuable results on longevity, maintenance effort, and robustness, especially in harsh environments. A disadvantage of a redundant communication system with global network coverage is its relatively high power consumption. The necessity of each individual communication stage should therefore be evaluated according to its particular application.

8.5 Control architecture

What does a flexible, modular and reliable software architecture for autonomous sailing boat control look like? How can existing sailing boat automation devices and methods be combined to allow boats to sail completely autonomously?

The multi-layer architecture presented combines deliberative and reactive methods and covers all the navigational tasks of the autonomous sailing boat. It consists of the following four layers:

- Strategic long-term routing
- Short course routing
- Manoeuvre execution
- Emergency reflexes

What is special about this approach is that all the layers are run asynchronously and in parallel. Each layer is only re-executed when this makes sense, for example because of new data becoming available. The results are then passed down to the next layer as a proposal. This next layer will normally then try to act on the proposal. Where however the detailed or updated data made available to the lower layer is in conflict with the prompts from the layer above, the lower layer may also ignore the prompts or only partially obey them.

The structure is modelled on the distribution of tasks amongst a human crew of sailors. Individual crew members can perform certain tasks by themselves and then pass the results to certain other crew members. For example, one person can be responsible for navigation and can plan a rough route on the basis of nautical charts and weather forecasts, and can pass this on to the helmsman. Generally the helmsman will follow the suggested routing. However, should local conditions arise of which the navigator was unaware (e.g. approaching severe weather, a local ban on sailing etc.) that make a different route more favourable, the helmsman can ignore the navigator's instructions and deviate from the proposed route.

This control architecture has been successfully tested on robotic sail boats of various sizes and in varying weather conditions. Particularly in the context of robotic sailing competitions, a direct comparison showed that the system can respond robustly and flexibly to rapidly changing conditions.

During development, the modular structure proved to be particularly advantageous:

- it was possible to swap individual layers (e.g. to test alternative algorithms for part sections)
- existing solutions could be easily incorporated as modules; no time synchronisation is necessary with other parts and the interfaces between the modules are very simple.

Some disadvantages were found owing to the complex construction with many tasks running in parallel:

- relatively powerful computing hardware is required, which leads to relatively high power consumption
- because of the parallel, asynchronous processing of the individual layers, software debugging can prove difficult.

8.6 Further work

The author currently works in a joint research group of the Austrian Society for Innovative Computer Sciences (INNOC) and Oregon State University (USA). The aim of the project is to use an autonomous sailing boat for passive acoustic monitoring of marine mammals and mitigation of human impacts on them. An autonomous sailing boat offers major advantages compared to submerged operated vehicles, including payload, speed, continuous real-time access, energy, and on-board computational power. However, there are also challenges which must be addressed:

- **Collision avoidance:** an important problem to be solved for long-term unmanned and autonomous missions at sea is to avoid collisions reliably. Although a reactive method for obstacle avoidance has been proposed and tested in computer simulation, there are some more tests to be done. Real-world tests, as well as experiments with moving and non-convex obstacles have not been done yet. While fixed obstacles such as landmasses can be predefined on the nautical chart which is the basis for the routing system, a wide range of obstacles need to be detected in real-time. Future research will be on a combination of multiple techniques, such as thermal imaging, radar, camera, and automatic identification system (AIS) to detect obstacles.

- **Energy balance:** the currently used ASV Roboat is showing a slightly negative energy balance. The solar system does not provide any spare power for the additional equipment for applications (e.g. sensors for environmental monitoring, search and rescue operations, etc.). In order to compensate this lack of energy, there are basically two possible approaches: generating more power or increasing efficiency. Saving power can be obtained by the use of more efficient components (computer, sensors, drives) and by optimising the control algorithms. Furthermore, a balanced rig design can help to reduce power consumption of the sail drive.

See Appendix B for a more detailed project description. Apart from this particular project, further research in autonomous sailing could focus on:

- **Adaptive control:** the proposed control algorithms are highly independent upon a specific boat type. However, a few parameters need to be adjusted in order to use the control software on another boat. A vision is to improve the software in a way that the boat automatically adjusts these parameters based on its own sailing experiences.

- **Swarm robotics:** once single robotic sailing boats work reliably and self-sufficiently in terms of power, swarm robotics technologies can be applied. Fleets of mass produced and affordable robotic sailing boats will cover the ocean surfaces and collaboratively undertake missions like measuring meteorological data, detect water pollution, rescue refugees and many more.

Robotic sailing boats are considered for a wide range of applications. A serious obstacle to successful implementation on a grand scale could be their unclear legal status. The International Regulations for Prevention of Collision at Sea (COLREGS) do not directly address unmanned autonomous surface vehicles (ASV). Nevertheless, some effort has been spent on implementation of algorithms to follow the COLREGS. However, it is not clear what type of vessel an ASV is according to the international maritime regulations. It seems, that the maritime laws need to be re-examined at least concerning vehicle classification. Once the classification is clear it then becomes difficult to identify other vessels reliably and to react properly.

Appendix A

CHR-based long-term routeing

Langbein, J.; Stelzer, R.; Frühwirth, T. (2011): A Rule-based Approach to Long-term Routing for autonomous Sailboats, in Proceedings of International Robotic Sailing Conference, Lübeck, DE.

Abstract – *We present an algorithm for long-term routing of autonomous sailboats with an application to the ASV Roboat. It is based on the A*-algorithm and incorporates changing weather conditions by dynamically adapting the underlying routing graph. We implemented our algorithm in the declarative rule-based programing language Constraint Handling Rules (CHR). A comparison with existing commercial applications yields considerably shorter computation times for our implementation. It works with real-life wind forecasts, takes individual parameters of the sailboat into account, and provides a graphical user interface.*

A Rule-based Approach to Long-term Routing for Autonomous Sailboats

Johannes Langbein, Roland Stelzer, and Thom Frühwirth

Abstract We present an algorithm for long-term routing of autonomous sailboats with an application to the ASV Roboat. It is based on the A*-algorithm and incorporates changing weather conditions by dynamically adapting the underlying routing graph. We implemented our algorithm in the declarative rule-based programing language Constraint Handling Rules (CHR) [4]. A comparison with existing commercial applications yields considerably shorter computation times for our implementation. It works with real-life wind forecasts, takes individual parameters of the sailboat into account, and provides a graphical user interface.

1 Introduction

Autonomous sailboats perform the complex maneuvers of sailing fully automatically and without human assistance. Starting off by calculating the best route based on weather data and going on to independent tacking and jibing, autonomous sailboats are able to sail through to any destination. Humans merely have to enter the destination coordinates.

The approach described here is planned to be implemented in the control system of the ASV Roboat, an autonomous sailing boat which has been in development by

Johannes Langbein
Faculty of Engineering and Computer Science, Ulm University, Germany e-mail: firstname.lastname@uni-ulm.de

Roland Stelzer
INNOC - Austrian Society for Innovative Computer Sciences, Vienna, Austria e-mail: firstname.lastname@innoc.at

Thom Frühwirth
Faculty of Engineering and Computer Science, Ulm University, Germany e-mail: firstname.lastname@uni-ulm.de

a research team of the Austrian Society for Innovative Computer Sciences (INNOC) since 2006.

So far, weather routing on the ASV Roboat relies on locally measured weather data only. This is proven to be suitable for short distances, respectively short durations, such as regattas over a few miles [12]. In contrast, for long-term missions like ocean crossings, weather conditions cannot be assumed to remain stable until the boat reaches its target. Therefore, a global view with consideration of weather forecasts is necessary.

In this paper, we introduce a long-term weather routing algorithm for autonomous sailboats and show how rule-based programming facilitates a declarative and efficient implementation. In Section 2 we present how we modeled the sailboat routing problem and introduce our routing algorithm. Its implementation is then discussed in Section 3. In Section 3.1, we compare our algorithm to existing commercial solutions and discuss related work. We conclude in Section 4, which also gives an outlook on future work.

2 The routing algorithm

Routing for sailboats, no matter whether they are autonomous or not, can be defined as the *"procedure, where an optimum track is determined for a particular vessel on a particular run, based on expected weather, sea state and ocean currents"* [11]. In this section, we will take a closer look at the parameters required to find an optimum track and present our routing algorithm.

2.1 Modeling long-term sailboat routes

For this work, we distinguish between long-term and short-term routing in the following way: Long-term routing is the task of finding a sequence of waypoints $\mathbf{x}_0 \dots \mathbf{x}_n$ (longitude and latitude coordinates) for a given starting point $\mathbf{x}_{start}$ and time t_{start} and a given destination point $\mathbf{x}_{dest}$, where $\mathbf{x}_{start} = \mathbf{x}_0$, $\mathbf{x}_{dest} = \mathbf{x}_n$ and $\mathbf{x}_k$ is reachable from $\mathbf{x}_{k-1}$ at sea while taking global weather forecasts into account. Short-term routing, in contrary, is the task of finding suitable boat headings to reach the next waypoint, given the current local weather conditions [12].

Several definitions for the term "optimum track" in the quote above are possible (see [12]), yet we want to focus on minimizing the arrival time t_{dest} at the point $\mathbf{x}_{dest}$. In order to calculate the arrival time t_k at any waypoint $\mathbf{x}_k$, we need to take weather data as well as the individual behavior of the sailboat into account.

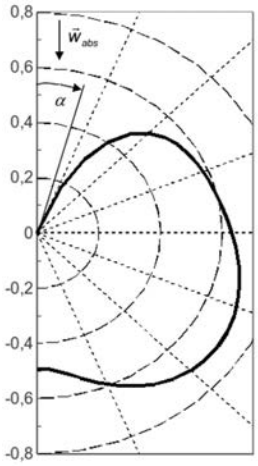


Fig. 1 The normalized polar diagram of the ASV Roboat

2.1.1 Weather data

As long-term routing is typically concerned with distances taking several days or weeks to travel, weather forecasts are required to calculate an optimal route. Weather forecasts are usually made available in the form of GRIB (Gridded Binary) files, a standardized format to store weather data [14]. In a GRIB file, wind conditions are represented as a grid of wind vectors $\mathbf{w}$, containing the wind-speed in north and east direction. A GRIB file can contain multiple forecasts, which are made available for up to 16 days in intervals as small as three hours. The resolution of the wind data typically ranges between 0.5 and 2.5 degrees.

2.1.2 Sailboat behavior

To calculate the time required to travel between two waypoints, we need to know the speed of the sailboat for given wind conditions. This speed can be described as a function of the wind-speed and the angle between the wind and the boats heading, that is to say, the boats velocity $v = \mathrm{v}(\mathbf{w}, \alpha)$, if α denotes the true wind angle of the boat. This function is usually shown in a plot known as *polar diagram*. Figure 1 shows the normalized polar diagram of the ASV Roboat [12], which describes the relation between wind-speed and boat-speed for a given true wind angle. Another factor to take into consideration is the so-called *hull-speed*, which we treat as the approximative maximum speed of the boat. In our algorithm, this maximum speed is configurable by the user.

We approximate the travel time t_{ij} between two locations $\mathbf{x}_i$ and $\mathbf{x}_j$ on a great circle path by taking the wind conditions $\mathbf{w}_i$ and true wind angle α_i at $\mathbf{x}_i$ for the first

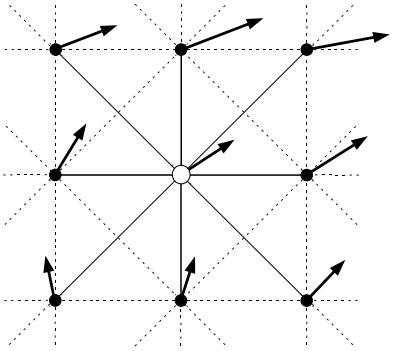


Fig. 2 An exemplary part of the routing graph. Wind vectors for each node are shown as bold arrows, edges to and from the center node as solid lines. Edges connecting the surrounding nodes are denoted as dashed lines.

half of the leg and the wind conditions $\mathbf{w}_j$ and true wind angle α_j at $\mathbf{x}_j$ for the second half. The distance d_{ij} between $\mathbf{x}_i$ and $\mathbf{x}_j$ is calculated using the laws of spherical geometry [2]. Together, we get the travel time $t_{ij} = \frac{1}{2} \cdot d_{ij} \cdot (v(\mathbf{w}_i, \alpha_i) + v(\mathbf{w}_j, \alpha_j))$. As sailing directly upwind is not possible, sometimes it is required to beat in order to sail from $\mathbf{x}_i$ to $\mathbf{x}_j$. We incorporate this into the travel time calculation by approximating the boats velocity made good along the great circle path between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the following way: As described in [7], we neglect the time for tacking and compute the velocity made good by using the convex hull of the polar diagram for speed calculations when sailing upwind. The true wind angle at which the boat is required to beat can be configured by the user.

2.1.3 Routing graph

Oceans constitute a continuous search space with an infinite number of possible waypoints. To reduce the search space and make classical shortest-path methods applicable to the routing problem, we chose to discretize the search space into a grid graph with equidistant nodes, representing points on the sea. Each node is connected to its eight nearest neighbors by directed edges. Nodes located on a land mass, which are detected using a binarized world map, are not included in the graph. Nodes at locations with hazardous wind conditions, that is to say, locations, at which the wind-speed is above a user-configurable limit, are omitted as well. Figure 2 illustrates a portion of such a routing graph.

Each node in the graph is annotated with wind vectors, shown as bold arrows in the figure. In most cases, we have multiple forecasts at hand, therefore each node is annotated with one wind vector per forecast. As the location of a node usually does not coincide with a grid point in the GRIB file, bi-linear interpolation is used to calculate the wind vectors for each node. The directed edges of the graph are annotated

with the travel time, calculated as shown in Section 2.1.2. Again, when multiple forecasts are present, each edge is annotated with one travel time per forecast.

The described discretization of the search space means, that the optimal route in the grid model is calculated, which is only an approximation to the true optimal route. Thus, we made the quality of the approximation configurable to the user in that the distance of the nodes in the routing graph can be arbitrarily chosen.

2.2 Calculating the optimal route

We chose the A*-algorithm as basis for our long-term routing as it allows the use of a heuristics for performance gain [5]. Much like Dijkstra's algorithm, the A*-algorithm assigns each node $\mathbf{x}_i$ a cost-value $\mathrm{c}(\mathbf{x}_i)$, which is the time required to travel to this node from the starting point. The algorithm retains an *open list* from which the currently best node is chosen for expansion. In contrary to Dijkstra's algorithm, the best node is not the node having the lowest cost but the node having the lowest value $\mathrm{c}(\mathbf{x}_i) + \mathrm{h}(\mathbf{x}_i)$, where $\mathrm{h}(\mathbf{x}_i)$ is the value of the heuristics for the node $\mathbf{x}_i$. This heuristics gives an estimate for the cost to reach the node $\mathbf{x}_{dest}$ from $\mathbf{x}_i$. We use the distance from $\mathbf{x}_i$ to $\mathbf{x}_{dest}$ divided by the boats hull-speed as heuristic function $\mathrm{h}(\mathbf{x}_i)$. It is *consistent* as it satisfies the triangle inequality $\mathrm{h}(\mathbf{x}_i) \leq t_{ij} + \mathrm{h}(\mathbf{x}_j)$ for two adjacent nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ and also the condition $\mathrm{h}(\mathbf{x}_{dest}) = 0$. We choose this heuristics as it is cheap to compute and its consistency guarantees the optimality of the A*-algorithm [5].

To find an optimal route, the A*-algorithm is run on the routing graph described in Section 2.1.3. As the number of nodes and edges in the routing graph can get very high, we tried to optimize the graph construction to save memory and computation time in the following way:

Some nodes in the open list do not need to be expanded, if the destination node is reached before they are considered. Also, the fact that our heuristics is consistent guarantees that a node will never be expanded more than once by the A*-algorithm [5]. This allows to dynamically construct and deconstruct the routing graph during the execution of the algorithm:

- A node $\mathbf{x}_j$ is not created until one of the eight direct neighbors $\mathbf{x}_i$ in the grid is chosen for expansion.
- Edges are not created until two adjacent nodes are present.
- If a node is added to the closed list, every incoming and outgoing edge to and from this node can be removed from the graph, as nodes in the closed list will not be considered again.

The dynamic construction of the graph also facilitates another optimization: When a node $\mathbf{x}_i$ is expanded, the time $t_{start} + \mathrm{c}(\mathbf{x}_i)$, at which this node will be reached by the boat, is known. Consequently, the forecast in the GRIB file valid at this point in time is known and the calculation of the travel time for outgoing edges of $\mathbf{x}_i$ has to be done only for this forecast and not for all forecasts in the GRIB file.

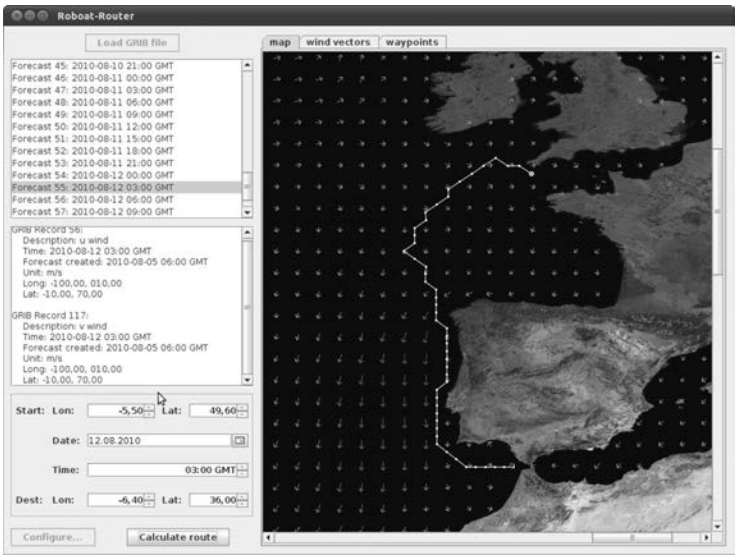


Fig. 3 The GUI showing wind conditions and the calculated route

3 Implementation in CHR

We implemented our algorithm mainly in Constraint Handling Rules (CHR), combined with SWI-Prolog. For an introduction to CHR, we refer to [4]. Our implementation is based on an existing implementation of Dijkstra's algorithm with a Fibonacci heap in CHR [10], which is used as open list to achieve optimal time complexity and was extended to incorporated the use of the heuristic function. Our implementation uses CHR rules for the following tasks:

- Expanding a node: Every time a node is extracted from the open list, a rule is triggering the creation of the neighboring nodes, if they are not yet present.
- Creating edges: If there are two neighboring nodes, a rule creates an edge between the two nodes with the according travel time.
- Labeling neighbor nodes: As soon as edges are present, a rule labels the neighbors of the node expanded last with the according cost and inserts them into the open list.
- Adding node to closed list: If there are no more neighbors to be labeled, a node is added to the closed list.
- Removing edges: A node in the closed list triggers a rule removing all the incoming and outgoing edges of this node.
- Path reconstruction: Once the goal node is reached, a CHR rule reconstructs the shortest path found.

We chose CHR for its declarativity, which allowed us to implement the routing algorithm in a compact and clear fashion: The implementation in CHR consists of

only 17 rules with a little under 1000 lines of code. In addition, CHR allows the routing graph to be constructed and deconstructed dynamically by simply stating the conditions, under which nodes and edges are created or removed, as rules. We furthermore believe, that the implementation in CHR facilitates future adaption and extensions in an easy way, which will help to incorporate changes that might be necessary on the ground of evaluations to come in real-life settings and conditions.

Our implementation can be used from the Prolog command line or via a Java application providing a graphical user interface. The GUI visualizes the wind conditions and allows to define starting and destination points on a world map. It also provides configuration options for the routing and displays the calculated route on the map. Figure 3 shows the GUI application.

3.1 Evaluation and Related Work

There are various commercial applications for long-term weather routing of sailboats like the *BonVoyage System*, *MaxSea*, *Sailplanner* [9], or *SailFast* [8]. The latter two are available as demo version, thus we picked them to compare our routing algorithm to. Both applications offer a GUI similar to ours. While Sailplanner automatically downloads its own wind data and is restricted to one provider (WeatherTech), SailFast allows the usage of arbitrary GRIB files. Sailplanner offers the user to pick from five different resolutions for the routing graph it uses, while SailFast uses an isochron method with an unknown resolution but with configurable time steps. However, they are both closed source applications, not revealing details about the algorithm they use for the routing.

We picked two routes of equal distance between starting point and destination (about 1440 km on a great circle path), one along the east-coast of the U.S. and one in open water to compare the running time of the two commercial solutions to our algorithm. The tests were carried out on a 2.0 GHz Intel Dual Core with 4 GB of RAM while no other applications where running and the wall clock time taken for the computation was measured. The output of Sailplanner indicates a grid width of about 30 km and 20 km for the resolutions “Medium High” and “Ultra High”, respectively. Hence, we chose 30 km and 20 km as the grid resolution for the runs of our algorithm. SailFast is fixed to a 6 hour resolution for the isochron lines in the demo version. The results of the comparison are given in Table 1.

The results show, that our algorithm is considerably faster when calculating routes. A reason for this the could be the use of a heuristic function in our algorithm or the fact, that SailFast and Sailplanner seem to consider points on land in the routing as well, while our algorithm avoids them. However, the most likely reason is the fact, that more than eight different bearings for each point are considered by SailFast and Sailplanner in contrary to our algorithm. The results in Table 1 also indicate the expected trade-off between computation time and quality of the approximation to the optimal route, stemming from the complexity of the A*-algorithm which is

Table 1 Comparison of wall clock computation time required for routing in seconds

	SailFast 6 hours	Sailplanner "med. high"	Sailplanner "ultra high"	Roboat router 20 km	Roboat router 30 km
Route 1	109	40	254	10	6
Route 2	79	37	224	9	5

exponential in the number of way points in the solution and thus the resolution of the grid [5].

The routes computed by our algorithm and SailFast are almost identical, while the route computed by Sailplanner is somewhat different. There are two reasons this difference might stem from: Firstly, Sailplanner uses a different polar diagram, which could not be changed in the demo version available to us. Secondly, Sailplanner was run with wind data from WeatherTech, as it does not allow the import of GRIB files. Our algorithm and SailFast were both run with the same GRIB file from saildocs.com since the data from WeatherTech is not freely available.

In academic research, several methods have been proposed for sailboat routing (see [12] for an overview). A stochastic method for long-term routing based on dynamic programming was presented by Philpott and Allsopp [7], while methods from operations research were used by Papadakis and Perakis [6]. Recent work by the AVALON team [3] uses a routing algorithm similar to ours, however, they do not include weather forecasts in their calculations. To our knowledge, none of the aforementioned approaches have been implemented in a rule-based language like our algorithm and there has not been a published implementation of a long-term weather routing algorithm for sailboats in a declarative language.

A popular algorithm for path planning in continuous search spaces is the Theta*-algorithm [1] which also works on a grid of nodes. It allows for any-angle path planning, creating edges to all nodes in sight of the node currently expanded. An implementation of this algorithm would provide for a more accurate routing as more angles are considered. However, this would require many more nodes to be held in memory, lead to a higher calculation time, and would raise difficulties at choosing the appropriate forecasts for calculating the travel time of an edge. Hence, we chose the A*-algorithm with dynamic construction of the routing graph over Theta*.

The D*-algorithm [13] and its variants are designed for searching in dynamically changing graphs. Their advantage is the fast replanning when costs of edges are modified during the execution of the path, which is particularly interesting if a robot continuously collects new information about its environment along the route. In robotic sailing however, costs are only updated when new forecasts are available, which usually happens only every couple of hours. Furthermore, the D*-algorithm reconsiders nodes in the closed list, which would infer with the dynamic construction and deconstruction of the our routing graph (cf. Section 2.2). For those reasons, we opted against the D*-algorithm in favor of a much simpler implementation and less memory consumption, while accepting the necessity of a full re-routing once new forecasts are available.

4 Conclusion and Future Work

We proposed a long-term routing algorithm which finds an arbitrarily accurate approximation to the optimal route for a sailboat for real-life wind conditions. Our approach takes changing weather conditions into account by dynamically adapting the underlying graph used as input to the A*-algorithm. It is is implemented in Constraint Handling Rules and compares very well to existing solutions. To our knowledge, it is the first published implementation of an long-term weather routing algorithm for sailboats in a declarative or rule-based programming language.

4.1 Future Work

We use deterministic weather forecasts which are only available for a certain time ahead (see Section 2.1.1). So-called *ensemble forecasts* consist of different scenarios that can happen with given probabilities and are usually available for a longer time ahead. Research for sailing yacht races has shown how ensemble forecasts can be used to find optimal routes which perform well under all possible scenarios [7]. An extension of the routing algorithm to handle ensemble forecasts could make the routing more realistic and would allow for accurate planning of even longer routes.

Besides wind conditions, the travel time of the sailboat can be affected by currents as well. Incorporating ocean current data into the calculation of the travel time could also make the routing more accurate and may lead to better routes.

Using a graph where nodes are connected only to their eight direct neighbors puts a considerable restriction on possible paths and can lead to noticeable differences between projected and actual arrival time. This restriction could be lowered, while improving the quality of the calculated route, by using a graph where nodes are connected to 24 neighbors, as presented in [3]. We believe that this could be achieved with a reasonable increase in calculation time by replacing the CHR rules for node and edge generation.

References

1. Daniel, K., Nash, A., Koenig, S., Felner, A.: Theta*: Any-Angle Path Planning on Grids. Journal of Artificial Intelligence Research **39**, 533–579 (2010)
2. Donnay, J.D.H.: Spherical Trigonometry. Interscience Publishers (2007)
3. Erckens, H., Büsser, G.A., Pradalier, C., Siegwart, R.Y.: Avalon: Navigation Strategy and Trajectory Following Controller for an Autonomous Sailing Vessel. IEEE Robotics & Automation magazine **17**(1), 45–54 (2010)
4. Frühwirth, T.: Constraint Handling Rules. Cambridge University Press (2009)
5. Hart, P., Nilsson, N., Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics **4**(2), 100 – 107 (1968)
6. Papadakis, N.A., Perakis, A.N.: Deterministic Minimal Time Vessel Routing. Operations Research **38**(3), 426–438 (1990)

7. Philpott, A., Mason, A.: Optimising Yacht Routes under Uncertainty. In: Proceedings of the 15th Chesapeake Sailing Yacht Symposium (CSYS 2000) (2000)
8. SailFast LLC: SailFast Version 5.1. http://www.sailfastllc.com/ (2011)
9. Sailport AB: Sailplanner. http://sailplanner.net/ (2011)
10. Sneyers, J., Schrijvers, T., Demoen, B.: Dijkstra's Algorithm with Fibonacci Heaps: An Executable Description in CHR. In: Proceedings of the 20th Workshop on Logic Programming (WLP 2006) (2006)
11. Spaans, J.A.: Windship routeing. Journal of Wind Engineering and Industrial Aerodynamics **19**, 215 – 250 (1985)
12. Stelzer, R., Pröll, T.: Autonomous Sailboat Navigation for Short Course Racing. Robotics and Autonomous Systems **56**(7), 604 – 614 (2008)
13. Stentz, A.: Optimal and efficient path planning for unknown and dynamic environments. International Journal of Robotics and Automation **10**(3), 89–100 (1993)
14. A Guide to the Code Form FM 92-IX Ext. GRIB Edition 1. http://www.wmo.int/pages/prog/www/WMOCodes/Guides/GRIB/GRIB1-Contents.html (1994). Accessed 11/28/2010

Appendix B

AAS Endurance: An autonomous acoustic sailboat for marine mammal research

Klinck, H.; Stelzer, R.; Jafarmadar, K.; Mellinger, D.K. (2007): AAS Endurance: An autonomous acoustic sailboat for marine mammal research, in Proceedings of International Robotic Sailing Conference, pp. 43-48, Matosinhos, Portugal.

Abstract *– This paper presents a joint research project of the Austrian Society for Innovative Computer Science, Austria and Oregon State University, USA which is intended to be realised within the next three years. The aim of the project is to develop an autonomous sailboat for passive acoustic monitoring of marine mammals and mitigation of human impacts on them. Performance tests of the autonomous acoustic sailboat - AAS Endurance - will include an open sea transect of at least one month duration. The work presented here discusses shortcomings of current ways of acoustic marine mammal monitoring and outlines advantages of a robotic sailboat for this task, as well as problems to be solved with this new technology.*

AAS Endurance: An autonomous acoustic sailboat for marine mammal research

Holger Klinck*, Roland Stelzer†‡, Karim Jafarmadar† and David K. Mellinger*
*Cooperative Institute for Marine Resources Studies
Oregon State University, Hatfield Marine Science Center
2030 SE Marine Science Drive, Newport, OR 97365, USA
Email: {holger.klinck,david.mellinger}@oregonstate.edu
†Austrian Society for Innovative Computer Science
Kampstraße 15/1, 1200 Vienna, Austria
Email: {roland.stelzer,karim.jafarmadar}@innoc.at
‡Center for Computational Intelligence
School of Computing at De Montfort University
The Gateway, Leicester LE1 9BH, United Kingdom

***Abstract*—This paper presents a joint research project of the Austrian Society for Innovative Computer Science, Austria and Oregon State University, USA which is intended to be realised within the next three years. The aim of the project is to develop an autonomous sailboat for passive acoustic monitoring of marine mammals and mitigation of human impacts on them. Performance tests of the autonomous acoustic sailboat - *AAS Endurance* - will include an open sea transect of at least one month duration. The work presented here discusses shortcomings of current ways of acoustic marine mammal monitoring and outlines advantages of a robotic sailboat for this task, as well as problems to be solved with this new technology.**

***Index Terms*—autonomous sailboat, robotics, marine mammals, bioacoustics, passive acoustic survey, underwater acoustics, line transect**

I. Introduction

Passive acoustic monitoring (PAM) is a widely used technique to estimate the abundance and distribution of marine mammals. A principal problem of PAM is the limitation in spatial and temporal coverage of the observations (see Fig. 1). Measurement can either be done with a moving platform (e.g., research vessel) or stationary recording devices (e.g., anchored autonomous recorders). Moving platforms offer the possibility of sampling a large area in a short period of time. However, because of the high costs of ship time, such passive acoustic line transects can be conducted only occasionally, and temporal coverage is very limited. In contrast, stationary recording devices [1] allow continuous sampling of an area. Their disadvantage lies in the limited spatial coverage of the devices.

Autonomous and remotely navigable passive acoustic platforms offer the possibility of sampling an area of interest with high temporal and spatial resolution at low cost. In this paper we introduce such a technology based on an autonomous acoustic sailboat (AAS). The extended payload and availability of energy on the proposed research platform allows operation of additional sensors such as measurement of chlorophyll and zooplankton density. The multi-sensor platform is therefore well-suited for investigating broader oceanographic and ecological questions, including predator-prey dynamics, patch scales, prey densities, and trophic energy flow.

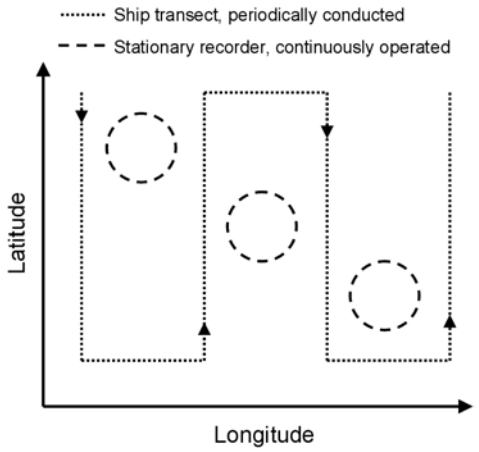


Fig. 1. Comparison of the spatial and temporal coverage of ship transects (dotted line) and stationary recorders (dashed circles)

II. Autonomous and Remotely Navigable Passive Acoustic Platforms for Marine Mammal Research

To date, two autonomous and remotely navigable passive acoustic platforms are available for marine mammal research: wave-powered vessels (e.g., the Wave Glider^{TM} [2]) and ocean gliders (e.g., the Seaglider^{TM} [3]).

The Wave Glider provides a submerged (swimmer) and a surface (float) unit. Both units are connected via a tether and allow the swimmer to move up and down as a result

of wave motion. The swimmer includes several fins which interact with the water as the swimmer moves up and down, and generate forces which propel the vehicle forward. The Wave Glider, developed by Liquid Robotics, Inc., has proven long-term capabilities in a five-month test trial, and the device seems well-suited for long-term passive acoustic monitoring of marine mammals. However, as the Wave Glider is a relatively new device, and to date there have been no reports of long-term acoustic recording capability, the following discussion will focus on the comparison of gliders with the proposed autonomous acoustic sailboat.

Gliders are commercially available from several manufacturers (e.g., [4]), and all types are based on the same principle. Changes in buoyancy cause the glider to move down and up in the water, and as with a airplane gliders, wings transform this vertical motion into forward motion. A stable, low-drag, hydrodynamic shape allows the glider to fly efficiently through the oceans. These devices are optimized for extremely low energy requirements and designed to operate at depths up to 1000 m. Gliders are capable of long-term operation and have been used extensively for oceanographic research for a number of years.

In the last years several research groups in the United States and Canada have started using gliders to investigate cetaceans [5], especially the deep-diving species in the beaked whale family (Ziphiidae; [6]). Research on beaked whales came to the fore because little is known about these animals and because of atypical stranding events which are suspected to be related to military sonar activities [7]–[9].

Because of their long dives (up to 1.5 h) and brief surfacing periods these animals are difficult to detect visually. Beaked whales vocalize extensively underwater to navigate and detect prey [10]. PAM is therefore the preferred method to determine presence/absence of beaked whales. However, as beaked whales appear to start echolocating at depths greater than 400 m, and because their emission beam pattern is narrow [11], the detection probability increases with depth [12] and sound reaches the surface only occasionally. Accordingly gliders are better suited for investigation of these animals than surface vessels.

Gliders are also used to investigate baleen whales [5]. Because of the glider's low speed (0.25-0.5 m/s, or 0.5-1 kt), flow noise is relatively low, which is advantageous for recording low-frequency baleen whale vocalizations. However, the internal electronics and mechanics of gliders periodically produce self-noise, and during such periods passive acoustic observations are not possible. An advantage of submerged operated vehicles is the limited surface time, which minimizes the risk of a collision with other obstacles, reduces damage from high-energy surface phenomena (wind and waves), and reduces the possibility of potentially harmful human action. Furthermore gliders can be deployed in polar regions, where ice coverage prohibits the usage of surface vehicles, and in areas with high wind and waves where the traditional visual means of marine mammal observation are ineffective.

III. Limitations of Passive Acoustic Gliders

Submerged operated platforms such as gliders also suffer from some drawbacks:

- **Speed:** The typical horizontal cruise speed of most gliders is approximately 0.25 m/s (0.5 kt). This low speed does not allow surveying a large area for a target species in a reasonably short time period. To be able to conduct a survey in a shorter amount of time, a larger number of gliders (number depending on the size of the area of interest) must be deployed. A larger number of devices significantly increases the complexity and cost of a survey.
- **Payload:** Most gliders are relatively small instruments and provide relatively limited payload capacity. Larger payloads allow for more batteries and sensors, so the small capacity of gliders limits both their deployment duration and their capability for measuring a wider suite of oceanographic parameters. An additional constraint in gliders is that the payload must be horizontally balanced.
- **Continuous real-time access:** As gliders stay submerged most of time, these platforms do not provide continuous real-time access. For real-time monitoring, such to warn of the presence of an endangered species, the minimum response time of a glider is the time it takes to rise to the surface - potentially several hours - plus a small amount of data transmission time.
- **Sensors:** The operating power for gliders comes from batteries. Because of constraints in payload mass, the amount of energy available for operating power-intensive electronics such as optical sensors is small.
- **Computational power:** Because of the energetic limitations, sophisticated and thus energy-intensive computations cannot be run continuously onboard a glider.
- **Reliability:** A malfunction at depth can cause the loss of a glider.
- **Duration:** Because of the limited energy capacity, acoustic glider deployments for marine mammal studies are limited to a duration of several weeks.

IV. Autonomous Sailing Vessels

An autonomous sailing vessel (ASV) is a sailboat equipped with sensors for wind speed and direction and motor-driven actuators for controlling sails, rudder, trim, etc. Using its intelligent control system [13]–[16], it can automatically steer the vessel to a desired point, maintain station at a location when desired, or follow any other long-term directions a shore-based pilot provides it. Autonomous sailboats are aimed to be used for several tasks on sea, especially for ocean sampling and observation [17]–[21].

The Roboat (see Fig. 2) is a type of ASV in development and use since 2007 [14], [15], [22], [23]. The basis for the Roboat is a commercial sailboat designed by Jan Herman Linge, the boat type Laerling. The boat was originally created for kids to learn sailing, and therefore safety and stability are the major characteristics of the boat. It has a length of 3.75 m

Fig. 2. The Roboat autonomous sailing vessel (ASV)

and comprises a 60 kg keel-ballast, which will bring the boat upright even from the most severe heeling. The boat can carry large payloads such as a battery bank and multiple sensors. Including batteries the overall weight of the boat is 300 kg. Additional payload of up to 50 kg is possible without impact on the sailing behaviour. The sail area of mainsail and foresail together is 4.5 m^2. It is equipped with solar panels providing up to 285 W of power during conditions of full sun and a direct methanol fuel cell delivering 65 W as a backup energy source. The Roboat features a three-stage communication system, combining WLAN, UMTS/GPRS and an IRIDIUM satellite communication system, allowing continuous real-time access from shore [23]. This can be used, for example, to track and navigate the ship, or to transmit information on acoustic detections, to a shore-based command center. The rudder and sails as well as the tacks and jibes are autonomously controlled by incoming data from various sensors (GPS, compass, anemometer, etc.) on an NMEA200-bus, which are analysed on an onboard PC running Linux. It has been successfully tested on Austrian Lakes, the Adriatic Sea in Croatia, and the Irish Sea in Wales. The Roboat is virtually unsinkable, so the danger of losing the device is small, and any detected system malfunctions can be immediately reported to the command center.

V. The AAS Endurance

The *AAS Endurance* will be a specially-equipped Roboat. Unique features of the *AAS Endurance* include the following.

A. Acoustic System

An acoustic streamer (towed array) will contain three hydrophones, a depth sensor, and a compass module for determining the orientation of the streamer. The captured sound will be sent to a BARIX Instreamer, which will digitize the analog signals with sampling rates up to 48 kHz. Data will be streamed continuously via the boat's WLAN interface to a base onshore, or to a manned vessel if within reach. This arrangement was successfully implemented and is being used in an autonomous listening station in Antarctica [24]. In parallel, the analog hydrophone signals will be sent to an onboard high-quality recording system with sampling rates up to 192 kHz and resolution of 24 b running on a low-power PC. Signals will also be sent to automated call-detection software running on DMON hardware developed by Mark Johnson of Woods Hole Oceanographic Institution. Such software will listen for calls of target species of marine mammals; such algorithms have been developed for many species of cetaceans (whales, dolphins, porpoises) and pinnipeds (seals, sea lions, walrus) (e.g., [25]–[28]). Most cetaceans and pinnipeds are reliably detectable from the surface, and data recorded from surface vessel towed arrays make clear that even beaked whales can be detected [29], although the detection probability is lower.

This acoustic data-capture and processing system will allow onboard real-time detection of marine mammal calls and storage of high-quality data for further laboratory analysis. If the sailboat's WLAN is within reach of shore, acoustic data can be streamed to the command center in real time. In addition, the spatially separated hydrophones provide information for estimating the direction to any sound sources encountered using time-of-arrival delay methods [30].

B. Optical System

An optical camera mounted at top of the mast can be aimed in any desired direction. The acoustic system will use its multiple hydrophones to estimate the bearing to a marine mammal sound source and provide this bearing to the optical system. The optical system can then be aimed in the desired direction to potentially allow visual identification of any vocalizing marine mammals when they surface.

C. Energy System

To produce energy independently of weather conditions, a methanol fuel cell is integrated as a backup system, allowing continuous provision of 65 W over a period of four weeks. The advanced energy system allows the Roboat to run sophisticated algorithms, such as for detection and classification of

marine mammal calls, continuously over extended periods of time. This energy system is not available on other types of autonomous acoustic platforms.

D. Speed

AAS Endurance will have a maximum speed of approximately 2.3 m/s (4.5 kt). This allows sampling an area of interest with high temporal and spatial resolution at low cost.

VI. Challenges

A. Obstacle Detection and Avoidance

An important problem to be solved for long-term unmanned and autonomous missions on sea is reliable obstacle detection and avoidance. Static obstacles such as landmasses can be predefined on the sea map which is the basis for the routing system. A combination of multiple techniques, such as thermal imaging, radar, camera, and automatic identification system (AIS) will be used to detect dynamic obstacles. Research in this field has been carried out for autonomous underwater vehicles [31] and motorised autonomous surface vehicles [32]–[35]. The obstacle avoidance task is different for sailing vessels, as they can not navigate in any direction directly, depending on wind conditions. Therefore a novel approach to autonomous obstacle avoidance will be an essential part of this research project.

B. Energy Balance

The currently used ASV Roboat can operate energetically autonomously with an average power consumption of 30 W. The solar system generates enough energy to sail continously, but doesn't provide any additional energy for the acoustic monitoring facilites. In order to compensate this lack of energy, there are basically two possible approaches: generating more power or increasing efficiency. The first approach within the research project will be to save power by the use of more efficient components (computer, sensors, drives) and by optimising the control algorithms. Furthermore, a balanced rig design (also known as Balestron rig, Aerorig^{TM}, swing rig, and EasyRig^{TM}) provides great potential to save power [36], [37]. A balanced rig consists of an unstayed mast carrying a main and jib (see Fig. 3). The main boom extends forward of the mast (the mast passes through the boom) to the tack of the jib. The main and jib are sized so that the force from the mainsail is slightly higher than that from the jib. That is, the combined center of effort is just behind the mast. Therefore the force needed to control the sheets is much lower than for a conventional sloop rig. The new rig will be equipped with motors for autmatic reefing in order to avoid damage during storms.

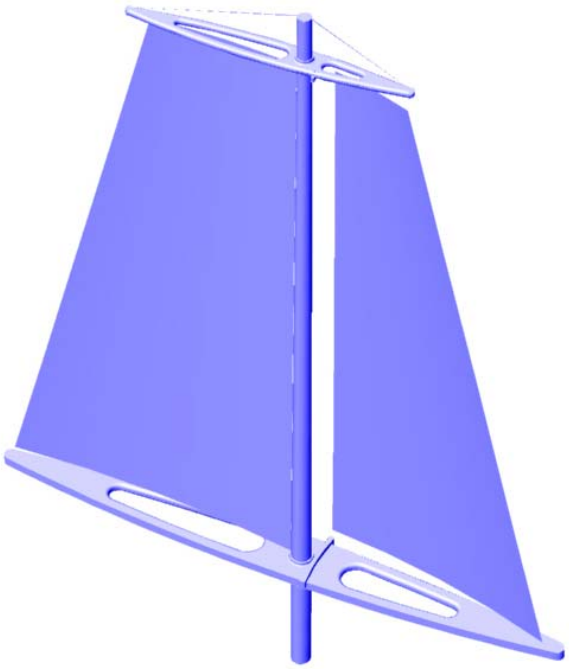


Fig. 3. Balanced rig example (source: [37])

VII. Project Timeline

To date (April 2009) the planning phase of the project is completed and funding has been requested. We plan to build and test *AAS Endurance* over the next three years.

In the first year of development the sailboat will be equipped with the control and energy system in Vienna, Austria. A first system test will be conducted on Lake Neusiedl, Austria. In a second step the acoustic system will be integrated. A more comprehensive test will be performed on the coast of the Baltic Sea in northern Germany. Goals of this test are (1) to verify that the control (including obstacle avoidance) and energy systems are working properly, (2) to evaluate the impact of the acoustic streamer on vessel speed and behavior, (3) to test mechanisms to optimize the depth and alignment of the acoustic streamer, and (4) to test the optical system for the potential verification of recorded sounds. A final tuning based on the result of the Baltic Sea test will be conducted in Vienna, Austria.

In the second year, *AAS Endurance* will undergo its first deep-water tests over 3-5 days off the coast of Newport, Oregon, USA. The goals of this test are optimization of the acoustic systems, especially noise reduction; assessment of vessel self-noise in various sea states; and testing of marine mammal detection capability. Some acoustic data will be transmitted in real time to shore, allowing analysis of acoustic system performance and wave and flow noise levels in various modes of sailing. Real-time marine mammal call detection algorithms will be implemented in the on-board acoustic system, allowing sending of encounter information nearly instantaneously via IRIDIUM communication link while on transect.

After successful completion of these tests, *AAS Endurance* will be transported to Hawaii, USA. After a final test off Kailua, Hawaii, USA, *AAS Endurance* will be sent on a transect from Kailua, Hawaii, USA to Newport, Oregon, USA, a direct distance of approximately 4100 km. The estimated

transect time is approximately 4 weeks. A comprehensive data analysis to characterize the system's performance at detecting marine mammal vocalizations will be conducted afterwards in the lab.

After the two-year development period, *AAS Endurance* will reach operational capability. A first scientific survey of marine mammals will be conducted in the third year.

VIII. Discussion and Future Work

The autonomous acoustic sailboat offers major advantages compared to submerged operated vehicles, including payload, speed, continuous real-time access, energy, and onboard computational power. However there are also challenges such as reliable obstacle avoidance linked to this new technology which must be addressed.

Gliders remain an important and powerful platform to investigate deep diving animals such as beaked whales or surveying polar regions where ice coverage prohibits the usage of surface vehicles. Both platforms are useful tools to gain knowledge of marine ecosystems, especially - as here proposed - of marine mammals.

AAS Endurance offers the operation of a multi-sensor platform and is therefore suitable to investigate broader ecological questions. The autonomous acoustic sailboat could, for example, be navigated to follow tagged animals using position information transmitted by the tag. Such a mission would help gain information on species-specific seasonal and diurnal vocalization in behavior. This baseline information is very important for projects utilizing passive acoustic recordings to estimate the distribution and abundance of marine mammals. Additional sensors for oceanographic variables such as chlorophyll and zooplankton density could help to understand the ecology of many marine mammal species.

References

[1] D. K. Mellinger, K. M. Stafford, S. E. Moore, R. P. Dziak, and H. Matsumoto, "An overview of fixed passive acoustic observation methods for cetaceans," *Oceanography*, vol. 20, no. 4, pp. 36–45, 2007.

[2] (2009). [Online]. Available: accessed on 9 April 2009, http://www.liquidr.com

[3] C. C. Eriksen, T. J. Osse, R. D. Light, T. Wen, T. W. Lehman, P. L. Sabin, J. W. Ballard, and A. M. Chiodi, "Seaglider: A long-range autonomous underwater vehicle for oceanographic research," *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 424–436, 2001.

[4] (2009). [Online]. Available: accessed on 9 April 2009, http://www.seaglider.washington.edu

[5] M. F. Baumgartner and D. M. Fratantoni, "Diel periodicity in both sei whale vocalization rates and the vertical migration of their copepod prey observed from ocean gliders," *Limnol. Oceanogr.*, vol. 53, no. 5, part 2, pp. 2197–2209, 2008.

[6] J. A. Theriault, D. Mosher, J. Hood, D. Flogeras, and T. Murphy, "Detection of beaked whales using autonomous underwater vehicle (glider)," Presentation with abstract, 3rd Intl. Workshop on Detection and Classification of Marine Mammals using Passive Acoustics, Boston, p. 24, 2007.

[7] A. Fernandez, J. . F. Edwards, F. Rodriguez, A. E. de los Monteros, P. Herraez, P. Castro, J. R. Jaber, V. Martin, and M. Arebelo, "Gas and fat embolic syndrome involving a mass stranding of beaked whales (family ziphiidae) exposed to anthropogenic sonar signals," *Veterinary Pathology*, vol. 42, pp. 446–457, 2005.

[8] T. M. Cox, T. J. Ragen, A. J. R. amd E. Vos, R. W. Baird, K. Balcomb, J. Barlow, J. Caldwell, T. Cranford, L. Crum, A. D'Amico, G. D'Spain, A. Fernandez, J. Finneran, R. Gentry, W. Gerth, F. Gulland, J. Hildebrand, D. Houser, T. Hullar, P. D. Jepson, D. Ketten, C. D. Macleod, P. Miller, S. Moore, D. C. Mountain, D. Palka, P. Ponganis, S. Rommel, T. Rowles, B. Taylor, P. Tyack, D. Wartzok, R. Gisiner, J. Mead, and L. Benner., "Understanding the impacts of anthropogenic sound on beaked whales," *J. Cetacean Res. Manage*, vol. 7, pp. 177–187, 2006.

[9] J. A. Hildebrand, "Impacts of anthropogenic sound," in *Marine Mammal Research: Conservation beyond Crisis*, J. E. R. III, W. F. Perrin, R. R. Reeves, S. Montgomery, and T. Ragen, Eds. Baltimore: Johns Hopkins Univ. Press, 2005, pp. 101–124.

[10] W. M. X. Zimmer, M. P. Johnson, P. T. Madsen, and P. L. Tyack, "Echolocation clicks of free-ranging cuvier's beaked whales (ziphius cavirostris)," *Journal of the Acoustical Society of America*, vol. 117, pp. 3919–3927, 2005.

[11] J. Ward, R. Morrissey, D. Moretti, N. DiMarzio, S. Jarvis, M. Johnson, P. Tyack, and C. White, "Passive acoustic detection and localization of mesoplodon densirostris (blainville's beaked whale) vocalizations using distributed bottom-mounted hydrophones in conjunction with a digital tag (dtag) recording," *Can. Acoust.*, vol. 36, pp. 60–66, 2008.

[12] W. M. X. Zimmer, J. Harwood, P. L. Tyack, M. P. Johnson, and P. T. Madsen, "Passive acoustic detection of deep-diving beaked whales," *Journal of the Acoustical Society of America*, vol. 124, pp. 2823–2832, 2008.

[13] J. C. Alves, T. M. Ramos, and N. A. Cruz, "A reconfigurable computing system for an autonomous sailboat," in *International Robotic Sailing Conference (IRSC)*. Breitenbrunn, Austria: Austrian Society for Innovative Computer Science, May 2008, pp. 13–20.

[14] R. Stelzer, T. Proell, and R. I. John, "Fuzzy logic control system for autonomous sailboats," in *FUZZ-IEEE*, London, UK, July 2007, pp. 97–102.

[15] R. Stelzer and T. Proell, "Autonomous sailboat navigation for short course racing," *Robotics and Auotnomous System*, vol. 56, no. 7, pp. 604–614, July 2008.

[16] M. L. V. Aartrijk, C. P. Taglioloa, and P. W. Adriaans, "Ai on the ocean: the robosail project," in *European Conference on Artificial Intelligence*, 2002, pp. 653–657).

[17] C. Sauze and M. Neal, "Design considerations for sailing robots performing long term autonomous oceanography," in *International Robotic Sailing Conference (IRSC)*. Breitenbrunn, Austria: Austrian Society for Innovative Computer Science, May 2008, pp. 21–29.

[18] N. A. Cruz and J. C. Alves, "Ocean sampling and surveillance using autonomous sailboats," in *International Robotic Sailing Conference (IRSC)*. Breitenbrunn, Austria: Austrian Society for Innovative Computer Science, May 2008, pp. 30–36.

[19] ——, "Autonomous sailboats: an emerging technology for ocean sampling and surveillance," in *MTS-IEEE Conference Oceans'2008*, Quebec, Canada, September 2008.

[20] C. Sauze and M. Neal, "An autonomous sailing robot for ocean observation," in *Towards Autonomous Robotic Systems (TAROS)*, Surrey, UK, 2006.

[21] M. Neal, "A hardware proof of concept of a sailing robot for ocean observation," *IEEE Journal of Ocean Engineering*, vol. 31, pp. 462–469, 2006.

[22] R. Stelzer and K. Jafarmadar, "A layered system architecture to control an autonomous sailboat," in *Towards Autonomous Robotic Systems (TAROS 2007)*, Aberystwyth, UK, September, September 2007, pp. 153–159.

[23] ——, "Communication architecture for autonomous sailboats," submitted to International Robotic Sailing Conference (IRSC), Porto, Portugal, July 2009.

[24] H. Klinck, "Automated passive acoustic detection, localization and identification of leopard seals: from hydro-acoustic technology to leopard seal ecology," *Reports on Polar and Marine Research*, vol. 582, 145 pp., 2008.

[25] D. K. Mellinger and C. W. Clark, "Recognizing transient low-frequency whale sounds by spectrogram correlation," *J. Acoust. Soc. Am.*, vol. 107, pp. 3518–3529, 2000.

[26] D. Gillespie, "Detection and classification of right whale calls using an edge detector operating on a smoothed spectrogram," *Can. Acoust.*, vol. 32, pp. 39–47, 2004.

[27] D. K. Mellinger, K. M. Stafford, and C. G. Fox, "Seasonal occurrence

of sperm whale (*Physeter macrocephalus*) sounds in the gulf of alaska, 1999-2001," *Mar. Mamm. Sci.*, vol. 20, no. 1, pp. 48–62, 2004.
[28] M. A. Roch, M. S. Soldevilla, R. Hoenigman, S. M. Wiggins, and J. A. Hildebrand, "Comparison of machine learning techniques for the classification of echolocation clicks from three species of odontocetes," *Can. Acoust.*, vol. 36, pp. 41–47, 2008.
[29] G. Pavan, C. Fossati, M. Priano, and M. Manghi, "Recording cuvier's beaked whales (*Ziphius cavirostris*) with a wideband towed array," Document SC/58/E18 submitted to the 58th International Whaling Commission Scientific Committee, 2008.
[30] W. A. Watkins and W. E. Schevill, "Spatial distribution of *Physeter catodon* (sperm whales) underwater," *Deep-Sea Res.*, vol. 24, pp. 693–699, 1977.
[31] S. Showalter, "The legal status of autonomous underwater vehicles," *Mar. Techn. Soc. J.*, vol. 38, no. 1, pp. 80–83, 2004.
[32] J. Larson, M. Bruch, R. Halterman, Rogers, and J. R. Webster, "Advances in autonomous obstacle avoidance for unmanned surface vehicles," in *Proceedings of AUVSI Unmanned Systems North America 2007*, vol. 582, Washington, DC, USA, 2007, pp. 154–168.
[33] M. R. Benjamin, J. J. Leonard, J. A. Curcio, and P. M. Newman, "A method for protocol-based collision avoidance between autonomous marine surface craft," *J. Field Robot.*, vol. 23, no. 5, pp. 333–346, 2006.
[34] T. Statheros, G. Howells, and K. McDonald-Maier, "Autonomous ship collision avoidance navigation concepts, technologies and techniques," *J. Navigation*, vol. 61, no. 1, pp. 129–142, 2008.
[35] R. Smierzchalski, "Evolutionary-fuzzy system of safe ship steering in a collision situation at sea," in *International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, vol. 1, 2005, pp. 893–898.
[36] (2009). [Online]. Available: accessed on 16 April 2009, http://www.multirig.com/the_balestron_rig.htm
[37] (2009). [Online]. Available: accessed on 16 April 2009, http://balancedrig.com/description.html

Appendix C

Sensor network developed for Roboat I

Stelzer, R.; Jafarmadar, K. (2007): Simple Communication Protocol for Rapid Robot Prototyping, in Proceedings of Humanoid and Service Robotics Conference", Kosice, SK.

Abstract *– Aim of the method presented in this paper is to provide a simple and unique communication protocol for a wide variety of sensors and actuators. All of the devices are equipped with a microcontroller in order to translate the proprietary device dependent protocol into the protocol presented here, named Simple Sensor Network (SSN). Due to the very simple commands to request data from a sensor respectively to send data to an actuator via RS232, it is an appropriate way for a novice to get in touch with robotics without the need of expert knowledge in electronics or programming. SSN makes a system easy to maintain, to adapt, and to extend.*

Simple Communication Protocol for Rapid Robot Prototyping

Roland Stelzer and Karim Jafarmadar

***Abstract*— Aim of the method presented in this paper is to provide a simple and unique communication protocol for a wide variety of sensors and actuators. All of the devices are equipped with a microcontroller in order to translate the proprietary device dependent protocol into the protocol presented here, named Simple Sensor Network (SSN). Due to the very simple commands to request data from a sensor respectively to send data to an actuator via RS232, it is an appropriate way for a novice to get in touch with robotics without the need of expert knowledge in electronics or programming. SSN makes a system easy to maintain, to adapt, and to extend.**

I. Introduction

Usually an autonomous robotic system consists of sensors, actuators and a computer controlling both device types and applying algorithms to them. Similar sensors and actuators often provide totally different interfaces. Therefore reimplementation and testing of interfaces requires a substantial part of development time. The proposed system provides a simple and unique communication protocol for a wide variety of sensors and actuators. The protocol is named Simple Sensor Network (SSN). A sensor or actuator in combination with a microcontroller forms an SSN module. Purpose of the microcontroller is to translate the proprietary device dependent protocol into SSN commands.

Once an SSN controller is developed for a certain sensor or actuator it can be reused in other applications by “plug and play”. The device network is scaleable in terms of numbers of devices and every device can be addressed individually. In contrast to common protocols used on embedded systems like I²C [1] or CAN [2] the proposed system is based on RS232 standard which makes it easily connectable to both personal computers and microcontrollers.

A light following SSN robot demonstrates the practicability and simplicity of the protocol. Furthermore SSN is implemented and tested successfully on the "Roboat" [3,4], a fully autonomous sailboat which won the Microtransat [5,6,7], an international competition in autonomous sailing.

II. Hardware Topology

An SSN topology consists of at least one SSN module – sensor or actuator – and a master. The master controls the entire system. The communication is always initiated by the master. An SSN module can directly be connected to the master’s serial port. In this case the number of SSN modules is limited to the number of serial ports of the master. If a PC is used as master system, usually not more than one or two serial interfaces are available.

To connect more SSN modules an additional device, the SSN switch, can be put in between the master and the SSN modules (Fig. 1). Up to 15 SSN modules can be connected to a single SSN switch. The SSN switch itself is connected to the master and listens for a request packet coming from the master. The SSN packet includes an address from 0x0 to 0xE, which specifies the port of the SSN switch to which the desired SSN Module is connected.

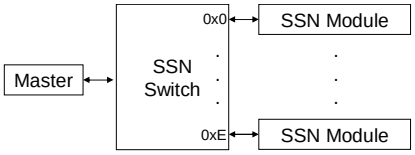


Fig. 1: SSN Topology

For systems with more than 15 SSN modules, SSN switches can be cascaded (Fig. 4, Fig. 5).

III. Communication Flow

The communication between master, SSN switch and SSN modules is always initiated by the master. The master can either set an actuators value or gather a sensor value. In both cases a request packet has to be sent from master to SSN switch. When a request occurs the SSN switch acts as follows:

1. SSN switch receives request packet from master.
2. SSN switch extracts the address part of the packet.
3. SSN switch forwards the complete packet to the switch-port corresponding to the address extracted before.
4. SSN switch waits for a reply packet from the SSN module and routes it back to the master. If no reply occurs within 100 ms, a timeout message will be returned to the master by the SSN switch.

A. Request Packet

The Request Packet consists of a header byte and an optional data part. The header is divided into four bits for the module address and four bits for the command, which has to be applied to the particular SSN module.

Roland Stelzer and Karim Jafarmadar are with the Austrian Association for Innovative Computer Science, Kampstraße 15/1, 1200 Vienna, Austria. E-Mail: roland.stelzer@innoc.at and karim.jafarmadar@innoc.at

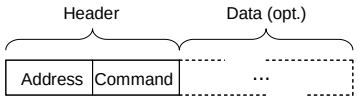


Fig. 2: Request Packet

Valid addresses are numbers from 0x0 to 0xE. 0xF indicates cascading of multiple SSN switches. The commands can be classified into three groups (Table I):

- Get general information about a particular SSN module (command id 0x0 – 0x1).
- Get value of a sensor device respectively status of an actuator (command id 0x2 – 0x5).
- Set value of an actuator device respectively change settings of a sensor (command id 0x6 – 0x9). These commands contain a data part in addition to the header. The length of data depends on the command.

TABLE I
SSN COMMANDS

Command ID	Command Description	Data length (bytes) request / reply
0x0	getinfo short	0 / 1
0x1	getinfo long (0-terminated)	0 / variable
0x2	getdata string (0-terminated)	0 / variable
0x3	getdata byte	0 / 1
0x4	getdata word	0 / 2
0x5	getdata double word	0 / 4
0x6	setdata string (0-terminated)	variable / 0
0x7	setdata byte	1 / 0
0x8	setdata word	2 / 0
0x9	setdata double word	4 / 0
0xA – 0xE	<not in use>	-
0xF	error indicator in reply packets	-

The commands *getinfo short* and *getinfo long* are supported by every SSN module. The reply packet on a *getinfo short* request gives information about the additionally supported commands. *getinfo long* returns a textual description of the particular SSN module.

B. Standard Reply Packet

When an SSN module receives a request packet, it processes the command and returns a reply packet. The first byte of the reply packet is identical to the first byte of the preceding request. Thus the most-significant half-byte contains the address (SSN switch port number where the module is connected to) and the least-significant half-byte describes the command. By means of the command, the SSN switch determines the number of bytes following the header byte (Table I).

C. Info Reply Packet

The commands *getinfo short* and *getinfo long* deliver general information about an SSN module. *getinfo long* returns a textual description of the SSN module as a 0-terminated string following the unmodified request header.

The reply packet on a *getinfo short* request gives information about the additionally supported commands.

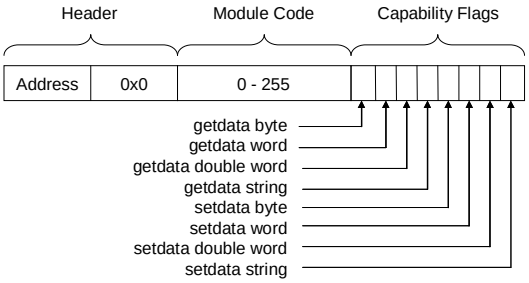


Fig. 3: Reply Packet for getinfo short

In addition to the unmodified request header the reply packet contains a module code, which can be used by the module developer to categorize SSN modules. A third byte informs about the range of implemented SSN commands (Fig. 3).

D. Timeout Reply Packet

If the SSN switch does not receive a reply packet from the SSN module within 100 ms a timeout occurs. In this case the SSN switch generates a timeout reply packet which consists of the address of the particular SSN module and the error indicator 0xF. No data bytes follow. A reply packet after timeout will be ignored.

IV. CASCADING

Through cascading of multiple SSN switches, systems with more than 15 SSN modules can be set up (Fig. 4). Each SSN switch can be equipped with a special cascade port with the cascade address 0xF.

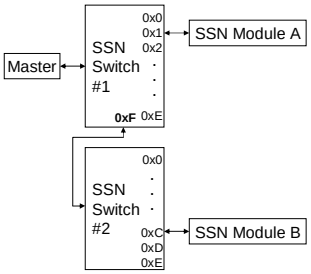


Fig. 4: Cascade Port

If more than one SSN switch has to be passed on the way between master and SSN Module, an extra header byte is necessary for each additional SSN switch. The most-significant half-byte of this header contains 0xF. This indicates that it is a cascade-header and the direct recipient of the packet is not an SSN module, but an additional SSN switch. The least-significant half-byte of the cascading header contains the port number to which the additional SSN switch is connected.

If the SSN switch receives a cascade header it discards the first byte. It forwards the rest of the packet to the following SSN switch and waits for a reply like with directly connected SSN modules. With this strategy, any number of SSN switches can be cascaded, as long as the packet runtime does not exceed the timeout limit.

Not only the standard cascade port 0xF, but any of the ports can be used to cascade (Fig. 5). By using multiple ports for cascading the number of hops between master and SSN module and the packet runtime can be minimized.

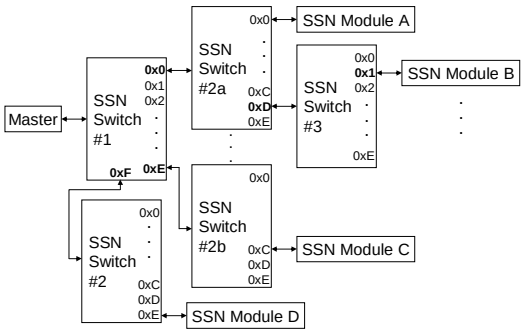


Fig. 5: Complex SSN Topology

V. Communication Example

The demonstration Topology is illustrated in Figure 5. SSN Module B is supposed to be a distance sensor. The command *getdata byte* (0x3) requests the measured distance in cm as 8 bit integer. The data flow from master to module is shown in Figure 6.

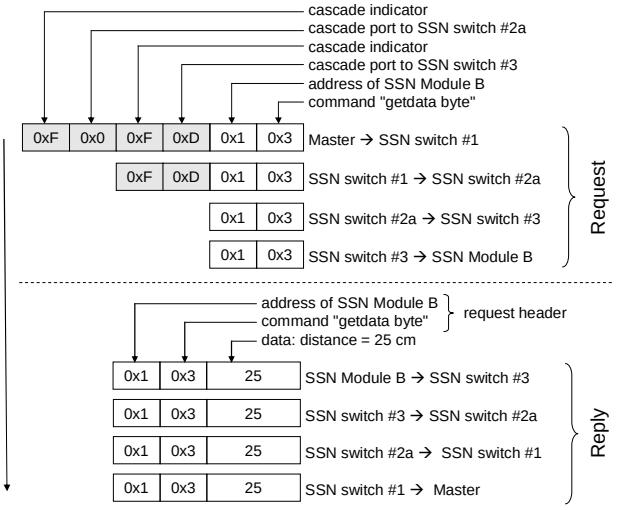


Fig. 6: Communication Example acc. to Fig. 5 - getdata byte from SSN Module B

VI. Application Example

For experiments controller hardware of SSN modules is based on Microchip PIC microcontrollers [8]. Software templates programmed in C [9] allow easy and fast development of new SSN modules.

A. Light Following SSN Robot

The robot in Figure 7 consists two side mounted independently driven wheels which are used for both propulsion and steering. This robot has a caster wheel mounted at the front to keep it from falling over. The robot is equipped with two brightness sensors, one pointed to the left, the other one to the right. The brightness sensors as well as the motor controllers for the wheels are connected to an SSN switch. Aim of the robot's algorithm is to follow a light source. The algorithm is executed by the master, which can be a laptop computer.

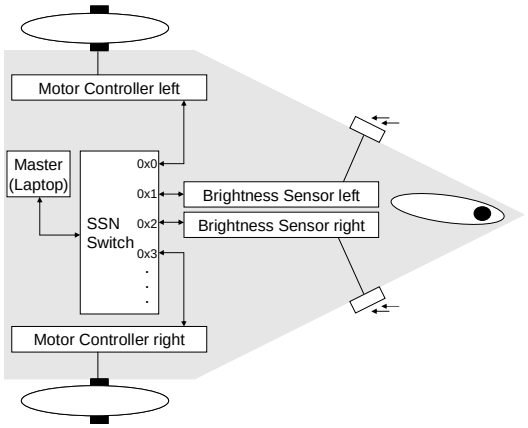


Fig. 7: Light Following SSN Robot

Only the SSN modules know about the device dependent protocols of the sensors and actuators and translate them to SSN commands. The master knows the SSN switch port number of every connected SSN module. The master communicates with the SSN modules using simple SSN commands (*setdata byte*, *getdata byte*). Therefore a few lines of code suffice to implement a light following robot.

Pseudo code example:

```
left=getdata_byte(0x1)   // read brightness left
right=getdata_byte(0x2)  // read brightness right
if ( left > right )      // light is left -> go to left
  setdata_byte(0x0,100)  // set left motor speed to 100
  setdata_byte(0x3,200)  // set right motor speed to 200
else                     // light is right -> go to right
  setdata_byte(0x0,200)  // set left motor speed to 200
  setdata_byte(0x3,100)  // set right motor speed to 100
```

B. Autonomous Sailboat "Roboat"

SSN is successfully implemented and used on the autonomous sailboat "Roboat". The boat won in the first Microtransat Challenge for autonomous sailboats in Toulouse, France, June 2006. The "Roboat" demonstrated completely autonomous sailing, where routeing, navigation and carrying out the manoeuvres run automatically and directly on the boat. For this purpose a sailboat is equipped with various sensors to measure the environmental conditions and actuators to control the rudder and sails. These devices are connected to the master via an SSN switch and communicate through SSN (Fig. 8).

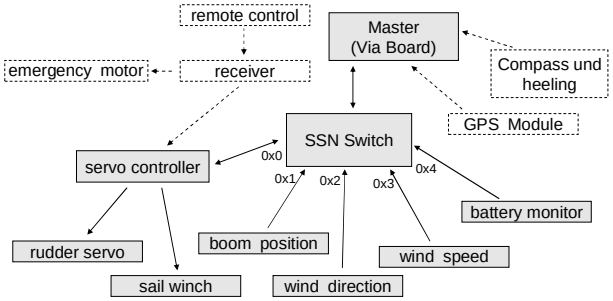


Fig. 8: "Roboat" Topology

VII. CONCLUSIONS

A robotic system applies an algorithm to a set of sensors and actuators. Different sensors respectively actuators usually mean different ways of communication. SSN provides a set of commands which allow uniform communication to a wide range of devices. Due to its simple command structure SSN is a proper way for novices to set up a robotic application. Once an SSN module is available there is no need to care about the specific characteristics of the sensor/actuator hardware.

The protocol is based on standardized serial communication (RS232). Therefore the SSN modules can be connected to a lot of different systems (PC, PDA, microcontrollers, etc.) and all popular programming languages can be used. Up to 15 SSN modules can be connected to a single SSN switch. Cascaded SSN switches allow larger installations.

The standardized set of SSN commands allows replacing sensors/actuators without changes in the master's program code. As an example, it is possible to replace an infrared distance sensor with an ultrasonic one.

SSN is successfully implemented and used on the autonomous sailboat "Roboat" and shows its strength especially in the prototyping phase where the system topology and the used devices frequently change. SSN makes a robot prototype easy to set up, to adapt, and to extend.

REFERENCES

[1] Philips Semiconductors, "The I²C-Bus Specifications", Version 2.1, Jan. 2000 [online]. Available: http://www.nxp.com/acrobat_download/literature/9398/39340011.pdf

[2] J.A. Gil, A. Pont, G. Benet, F.J. Blanes , M.Martínez "A CAN Architecture for an Intelligent Mobile Robot". *Proceedings of IFAC Symposium on Intelligent Components and Instruments for Control Applications SICICA'97*, pp.65-70. Annecy, 1997.

[3] R. Stelzer, T. Pröll, and R.I. John, Fuzzy Logic Control System for Autonomous Sailboats, Accepted for publication at *IEEE International Conference on Fuzzy Systems 2007*, London, UK.

[4] R. Stelzer and T. Pröll, "Autonomous sailboat navigation for short course racing," submitted for publication, 2007.

[5] Y. Briere, "First Microtransat Challenge" [online]. Available: www.ensica.fr/microtransat, 2006.

[6] C. Sauze and M. Neal, "An Autonomous Sailing Robot for Ocean Observation," in *Proceedings of TAROS Conference 2006* [online]. Available: http://taros.mech.surrey.ac.uk/papers/Sauze_Neal.pdf, 2006.

[7] M. Neal, "A hardware proof of concept of a sailing robot for ocean observation," *IEEE Journal of Ocean Engineering*, vol. 31, pp. 462-469, 2006.

[8] Microchip, "PIC Microcontrollers" [online]. Available: http://www.microchip.com

[9] Custom Computer Services Inc., "CCS C Compiler for PIC Microcontroller", [online]. Available: http://www.ccsinfo.com

References

Abril, J., Salom, J. & Calvo, O. (1997). Fuzzy control of a sailboat. *International Journal of Approximate Reasoning*, **16**, 359–375. 16, 17, 20, 28

Adriaans, P. (2003). From knowledge-based to skill-based systems: Sailing as a machine learning challenge. *Knowledge Discovery in Databases: PKDD 2003*, 1–8. 16

Airmar (2009). PB200 WeatherStation User Manual. Tech. rep., Airmar. xiii, 50

Alejos, A., Culnias, I., Sanchez, M. & Fernandez, J. (2007). Distress beacons for maritimal accidents: Measurements of coastal sea coverage. In *Antennas and Propagation, 2007. EuCAP 2007. The Second European Conference on*, 1–6, IET. 67

Allensworth, T. (1999). A short history of Sperry Marine. available online: `http://www.sperrymarine.northropgrumman.com/Company-Information/Corporate-History/Sperry-History/`. 15

Alves, J., Ramos, T. & Cruz, N. (2008). A reconfigurable computing system for an autonomous sailboat. In *IRSC 2008 International Robotic Sailing Conference*, 13–20. 39

Ammann, N., Hartmann, F., Jauer, P., Bruder, R. & Schlaefer, A. (2010). Design of a robotic sailboat for WRSC/SailBot. In *International Robotic Sailing Conference 2010*, 40–42. 38

Anschütz-Kaempfe, H. & von Shirach, F. (1904). Patent: Kreiselapparat (gyroscope). Deutsches Reichspatent no. 182855 (filed: 27 March 1904 ; issued: 2 April 1907). 15

Arkin, R. (1992). Homeostatic control for a mobile robot: Dynamic replanning in hazardous environments. *Journal of Robotic Systems*, **9**, 197–214. 25

Assilian, S. & Mamdani, E. (1974). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, **7**, 1–13. 8, 112

BalancedRig (2009). BalancedRig website. available online: `http://balancedrig.com/description.html` (accessed: 16 April 2009). viii, 18, 19, 55

Benatar, N., Qadir, O., Owen, J., Baxter, P. & Neal, M. (2009). P-Controller as an Expert System for Manoeuvring Rudderless Sail Boats. In *Proceedings of UKCI 2009*. viii, 16, 17

Benjamin, M., Leonard, J., Curcio, J. & Newman, P. (2006). A method for protocol-based collision avoidance between autonomous marine surface craft. *Journal of Field Robotics*, **23**, 333–346. 7, 25, 98

Bennett, S. (1986). *A history of control engineering, 1800-1930*. Inspec/Iee. 13, 15

Bertram, V. (2008). Unmanned surface vehicles–a survey. *Skibsteknisk Selskab, Copenhagen, Denmark*. 3

Blidberg, D. (2001). The development of autonomous underwater vehicles (AUVs); a brief summary. In *IEEE ICRA*, vol. 4, Citeseer. 26

Bohnenberger, J.G.F. (1817). Beschreibung einer Maschine zur Erläuterung der Gesetze der Umdrehung der Erde um ihre Axe, und der Veränderung der Lage der letzteren. *Tübinger Blätter für Naturwissenschaften und Arzneikunde*, **3**, 72–83. 14

Bowditch, N. (2010). *The American Practical Navigator*. Paradise Cay Publications. 1

Briere, Y. (2008). Iboat: An autonomous robot for long-term offshore operation. In *Electrotechnical Conference, 2008. MELECON 2008. The 14th IEEE Mediterranean*, 323–329, IEEE. 19, 20, 29, 35, 139

Briere, Y., Bastianelli, F., Gagneul, M. & Cormerais, P. (2005). Challenge Microtransat. In *CETSIS 2005, Nancy, France*. 29

Brooks, R. (1986). A robust layered control system for a mobile robot. *Robotics and Automation, IEEE Journal of*, **2**, 14–23. 25, 59

Bruder, R., Stender, B. & Schlaefer, A. (2009). Model sailboats as a testbed for artificial intelligence methods. In *In Proceedings of the 2nd International Robotic Sailing Conference*, 37–42. 38

Buffalo (2009). Buffalo airstation high power turbo g high power wireless cable/dsl smart router whr-hp-g54. available online: `http://www.buffalotech.com/files/downloads/WHR-HP-G54DS.pdf` (accessed: 20 May 2009). 73

Burnie, M., ed. (2010). *Participant Package – World Robotic Sailing Championship 2010 and International Robotic Sailing Conference 2010*. Queen's University, Kingston, Ontario, Canada. 20, 36, 38, 139

Burns, R. (1995). The use of artificial neural networks for the intelligent optimal control of surface ships. *Oceanic Engineering, IEEE Journal of*, **20**, 65–72. 15, 16

Caccia, M. (2006). Autonomous Surface Craft: prototypes and basic research issues. In *Control and Automation, 2006. MED'06. 14th Mediterranean Conference on*, 1–6, IEEE. 26

Das, B., Ray, D. & Majumder, S. (2011). Experiments with a behavior-based robot. *Computer Networks and Information Technologies*, 133–141. 59

Elkaim, G. (2002). *System identification for precision control of a wingsailed GPS-guided catamaran*. Ph.D. thesis, Standford University. 27, 28

Elkaim, G. (2006). The Atlantis project: A GPS-guided wing-sailed autonomous catamaran. *Navigation*, **53**. 18, 28

Elkaim, G. & Boyce, L.C.O. (2007). Experimental Aerodynamic Performance of a SelfTrimming Wing-Sail for Autonomous Surface Vehicles. In *In: Proc. Of the IFAC Conference on Control Applications in Marine Systems. IFAC CAMS*, Citeseer. 20

Enab, Y. (1996). Intelligent controller design for the ship steering problem. In *Control Theory and Applications, IEE Proceedings-*, vol. 143(1), 17–24, IET. 16

Eriksen, C.C., Osse, T.J., Light, R.D., Wen, T., Lehman, T.W., Sabin, P.L., Ballard, J.W. & Chiodi, A.M. (2001). Seaglider: A long-range autonomous underwater vehicle for oceanographic research. *IEEE Journal of Oceanic Engineering*, **26**, 424–436. 3

Finanztest (2006). Geld verdienen auf dem Dach. *Finanztest*, **7**, 35. 55

Frühwirth, T. (2009). *Constraint Handling Rules*. Cambridge University Press. 23, 134

Giger, L., Wismer, S., Boehl, S., Büsser, G., Erckens, H., Weber, J., Moser, P., Schwizer, P., Pradalier, C. & Siegwart, R. (2009). Design and construction of the autonomous sailing vessel Avalon. In *Proc. 2nd Int. Robotic Sailing Conf*, 17–22. 19, 20, 23, 24, 36, 55, 139

Hart, P., Nilsson, N. & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *Systems Science and Cybernetics, IEEE Transactions on*, **4**, 100–107. 23

Huaweii (2009). System description of huawei e220 usb modem (huawei technologies co., ltd.). available online: `http://corporate.proximus.be/download/VMCManualHuaweiE220.pdf` (accessed: 20 May 2009). 73

Iridium (2003). Iridium satellite data services white paper. available online: `http://www.stratosglobal.com/~/media/SGlobal/Documents/`

factsheets/irid/irid_whitePaper_satelliteDataServices.ashx (accessed: 25 May 2011). 68

Iridium (2009). System iridium 9601 sbd transceiver datasheet. available online: http://www.iridium.com/about/press/pdf/Iridium9601r2.pdf (accessed: 20 May 2009). 73

Langbein, J., Stelzer, R. & Frühwirth, T. (2011). A rule-based approach to long-term routing for autonomous sailboats. *Robotic Sailing*, 195–204. 23, 134

Larson, J., Bruch, M., Halterman, R., Rogers, J. & Webster, R. (2007). Advances in autonomous obstacle avoidance for unmanned surface vehicles. In *Proceedings of AUVSI Unmanned Systems North America 2007*, vol. 582, 154–169, Washington, DC, USA. 7, 25, 98

Layne, J. & Passino, K. (1993). Fuzzy model reference learning control for cargo ship steering. *Control Systems Magazine, IEEE*, **13**, 23–34. 16

Lee, S., Kwon, K. & Joh, J. (2004). A fuzzy logic for autonomous navigation of marine vehicles satisfying colreg guidelines. *International Journal of Control Automation and Systems*, **2**, 171–181. 25

Liquid Robotics (2009). Liquid robotics web site. available online: http://www.liquidr.com (accessed on 9 April 2009). 3

Loibner, D. (1998). KLAR zur WENDE? klick. *Yacht Revue*, **3**. 28

Manley, J. (2008). Unmanned surface vehicles, 15 years of development. In *OCEANS 2008*, 1–4, IEEE. 26

Maretron (2005). Gps100 gps/waas antenna/receiver. Tech. rep., Airmar. xiii, 51

Maretron (2006). Ssc 200 solid state rate/gyro compass user's manual. Tech. rep., Airmar. xiii, 51

McMahon, M. & Rathburn, R. (2005). Measuring latency in iridium satellite constellation data services. *US Naval Academy Report no: A291464, June 2005*. 68

Medved, M. & Tekovic, A. (2004). Extended range functionality for gsm networks. In *Electronics in Marine, 2004. Proceedings Elmar 2004. 46th International Symposium*, 211–216, IEEE. 67

Mellinger, D.K., Stafford, K.M., Moore, S.E., Dziak, R.P. & Matsumoto, H. (2007). An overview of fixed passive acoustic observation methods for cetaceans. *Oceanography*, **20**, 36–45. 3

Microtransat (2011). Official microtransat web site. available online: `http://www.microtransat.org` (accessed on 27 April 2011). 29

Miller, P., Brooks, O. & Hamlet, M. (2009). Development of the USNA sailbots (ASV). In *International Robotic Sailing Conference 2009*. 37

Miller, P., Beal, B., Capron, C., Gawboy, R., Mallory, P., Ness, C. & Petrosik, R. (2010). Increasing performance and added capabilities of USNA sail-powered autonomous surface vessels (ASV). In *International Robotic Sailing Conference 2010*. 37

Minorsky, N. (1922). Directional stability of automatic steered bodies. *Journal of American Society of Naval Engineers*, **34(2)**, 280–309. 15

Morgan, S.L. (2006). Guide to Microsoft Excel for calculations, statistics, and plotting data. Tech. rep., Department of Chemistry & Biochemistry, The University of South Carolina, Columbia. 129

Motte, R. & Calvert, S. (1990). On the selection of discrete grid systems for on-board micro-based weather routeing. *Journal of Navigation*, **43**, 104–117. 23

Motte, R., Burns, R. & Calvert, S. (1988). An overview of current methods used in weather routeing. *Journal of Navigation*, **41**, 101–114. 22

Multirig (2009). Multirig website. available online: http://www.multirig.com/the_balestron_rig.htm (accessed: 16 April 2009). 18, 55

Neal, M. (2006). A hardware proof of concept of a sailing robot for ocean observation. *Oceanic Engineering, IEEE Journal of*, **31**, 462–469. 37

Neal, M., Sauze, C., Thomas, B. & Alves, J. (2009). Technologies for Autonomous Sailing: Wings and Wind Sensors. *proceedings of the 2nd IRSC, Matosinhos, Portugal, July 6th-12th*, 23–30. 18

Philpott, A. & Mason, A. (2001). Optimising yacht routes under uncertainty. In *Proceedings of the 15th Chesapeake Sailing Yacht Symposium, Annapolis, MD*, Citeseer. 23, 24

Polkinghorne, M., Roberts, G., Burns, R. & Winwood, D. (1995). The implementation of fixed rulebase fuzzy logic to the control of small surface ships. *Control Engineering Practice*, **3**, 321–328. 16

Roberts, G. (2008). Trends in marine control systems. *Annual reviews in control*, **32**, 263–269. 13, 14, 15

RoboCup (2011). Official RoboCup website: Objective. available online: http://www.robocup.org/about-robocup/objective/ (accessed: 1 May 2011). 28

RoboSail (2011). Official RoboSail web site. available online: http://www.robosail.com (accessed: 28 April 2011). 16

Rynne, P. & von Ellenrieder, K. (2009). Unmanned autonomous sailing: Current status and future role in sustained ocean observations. *Marine Technology Society Journal*, **43**, 21–30. 3

SailFast LLC (2011). SailFast Version 5.1. available online: http://www.sailfastllc.com/ (accessed: 22 July 2011). 134

Sailport AB (2011). Sailplanner. available online: http://sailplanner.net/ (accessed: 22 July 2011). 134

Sauzé, C. & Neal, M. (2008). Design considerations for sailing robots performing long term autonomous oceanography. In *Proceedings of The International Robotic Sailing Conference, 23rd-24th May*, 21–29. 37

Sauze, C. & Neal, M. (2010). A Raycast Approach to Collision Avoidance in Sailing Robots. In *Proceedings of 3rd International Robotic Sailing Conference*, 26–33. 26

Scanmar (2011). Classification of vane gears by course correcting system. available online: `http://www.selfsteer.com/windvanes101/classification.php` (accessed on 27 April 2011). viii, 14

Schieben, E.W. (1969). SKAMP – an amazing unmanned sailboat! *Ocean Industry*, 38–43. 27

Schrag, H. (2011). Jahreserträge aus Photovoltaik-Anlagen unserer Kunden. available online: `http://www.schrag-sonnenstrom.de/9.html` (accessed: 28 September 2011). 55

Seaglider (2009). Seaglider web site. available online: `http://www.seaglider.washington.edu` (accessed on 9 April 2009). 4

Shamos, M. & Hoey, D. (1976). Geometric intersection problems. In *Proc. 17th IEEE Symp. Foundations of Computer Science (FOCS '76)*, vol. 134, 208–215, doi:10.1109/SFCS.1976.16. 103

Showalter, S. (2004). The legal status of autonomous underwater vehicles. *Marine Technology Society Journal*, **38**, 80–83. 7, 25, 98

Shukla, P. & Ghosh, K. (2009). Revival of the Modern Wing Sails for the Propulsion of Commercial Ships. *International Journal of Civil and Environmental Engineering*, 75–80. 18

Sliwka, J., Reilhac, P., Leloup, R., Crepier, P., Malet, H., Sittaramane, P., Bars, F., Roncin, K., Aizier, B. & Jaulin, L. (2009). Autonomous robotic boat of ensieta. In *2nd International Robotic Sailing Conference, Matosinhos, Portugal*. 35

Smierzchalski, R. (2005). Evolutionary-fuzzy system of safe ship steering in a collision situation at sea. In *International Conference on Intelligent Agents, Web Technologies and Internet Commerce*, vol. 1, 893–898, IEEE Computer Society. 7, 23, 25, 98

Smierzchalski, R. & Michalewicz, Z. (2000). Modeling of ship trajectory in collision situations by an evolutionary algorithm. *Evolutionary Computation, IEEE Transactions on*, **4**, 227–241. 23

Smith, B. (1970). SKAMP – roboat boat with rigid sails patrols ocean beat. *Popular Science*, **196**, 70–72. 27

Smith, C. & Collins, D. (2002). *3G wireless networks*. McGraw-Hill Professional. 66

Spaans, J. (1985). Windship routeing. *Journal of wind engineering and industrial aerodynamics*, **19**, 215–250. 22

Spaans, J. & Stoter, P. (1995). New developments in ship weather routing. *Navigation*, **43**, 95–106. 22

Sperry, E. (1922). *Automatic steering*. Society of Naval Architects and Marine Engineers. 15

Spiegel (1998). Fliegender Schwarzwälder. *Der Spiegel*, **22**, 176–177. 28

Statheros, T., Howells, G. & Maier, K. (2008). Autonomous ship collision avoidance navigation concepts, technologies and techniques. *The Journal of Navigation*, **61**, 129–142. 7, 25, 98

Stawicki, K. & Smierzchalski, R. (2001). Methods of optimal ship routeing for weather perturbations. In *IFAC Conference on Control Applications in Marine System*, 101–106, Glasgow, UK. 23

Sybrandy, A. (2007). Remote data acquisition telemetry options. available online: `http://stf.ucsd.edu/presentations/2007-02%20STF%20-%20Sybrandy%20-%20Telemetry.pdf` (accessed: 20 May 2009), scripps Technical Forum, University San Diego. 68

Takagi, T. & Sugeno, M. (1985). Fuzzy identification of system and its applications to modelling and control. *IEEE Trans. Syst., Man, and Cyber*, **1**, 5. 112

Thornton, T. (1993). A review of weather routeing of sailboats. *Journal of Navigation*, **46**, 113–129. 22

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G. *et al.* (2007). Stanley: The robot that won the DARPA Grand Challenge. *The 2005 DARPA Grand Challenge*, 1–43. 28

Van Aartrijk, M., Tagliola, C. & Adriaans, P. (2002). AI on the Ocean: the RoboSail Project. In *ECAI*, 653–657, Citeseer. 16

Van Amerongen, J. (1984). Adaptive steering of ships–A model reference approach. *Automatica*, **20**, 3–14. 15

Velagic, J., Vukic, Z. & Omerdic, E. (2003). Adaptive fuzzy ship autopilot for track-keeping. *Control engineering practice*, **11**, 433–443. 16

von Alt, C. (2003). Autonomous underwater vehicles. In *Autonomous Underwater Lagrangian Platforms and Sensors Workshop*. 26

WMO94 (1994). A Guide to the Code Form FM 92-IX Ext. GRIB Edition 1. available online: `http://www.wmo.int/pages/prog/www/WMOCodes/Guides/GRIB/GRIB1-Contents.html` (accessed: 28 November 2010. 134

Worsley, P. (2011). Self Trimming/Self Tending Wingsails. available online: `http://www.ayrs.org/wingsails.pdf` (accessed on 8 May 2011). 20

Wylie, C., Romney, G., Evans, D. & Erdahl, A. (1967). Halftone perspective drawings by computer. In *Proc. AFIPS FJCC*, vol. 31, 49. 103

Yeh, E. & Bin, J. (1992). Fuzzy control for self-steering of a sailboat. In *Intelligent Control and Instrumentation, 1992. SICICI'92. Proceedings., Singapore International Conference on*, vol. 2, 1339–1344, IEEE. 16

Zirilli, A., Tiano, A., Roberts, G. & Sutton, R. (2002). Fuzzy course-keeping autopilot for ships. In *15th Triennial World Congress, Barcelona, Spain*. 16